Algorithmic performance of large-scale distributed networks
A spectral method approach

A Thesis
Presented to
The Academic Faculty

by

Christos Gkantsidis

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

College of Computing
Georgia Institute of Technology
May 2006

Algorithmic performance of large-scale distributed networks
A spectral method approach

Approved by:

Dr. Milena Mihail, Advisor
College of Computing
Georgia Institute of Technology

Dr. Michalis Faloutsos
Computer Science Department
University of California, Riverside

Dr. Mostafa Ammar
College of Computing
Georgia Institute of Technology

Dr. Ellen Zegura
College of Computing
Georgia Institute of Technology

Dr. Constantinos Dovrolis
College of Computing
Georgia Institute of Technology

Date Approved: December 06, 2005

To my parents

ACKNOWLEDGEMENTS

I want to begin by expressing my sincere gratitude to my thesis advisor, Dr. Milena Mihail, for her support and guidance during my PhD. Milena has been a great teacher, a great advisor, and a great friend. The turning point in my PhD was four years ago when (an extremely patient and enthusiastic) Milena taught me about spectral clustering; I couldn't have found a better teacher! That meeting changed completely the focus of my research. Working with Milena has been a great pleasure.

I have benefited tremendously by collaborating with Dr. Mostafa Ammar and Dr. Ellen Zegura. Their guidance and insightful comments throughout my graduate studies were extremely valuable and I thank them for their help.

I also want to thank Dr. Constantinos Dovrolis and Dr. Yannis Smaragdakis for their invaluable help and friendship during my graduate studies.

I am very honored that Dr. Michalis Faloutsos is a member of my thesis committee. His work on Internet topologies influenced significantly my own research and I am very happy that he agreed to serve in my committee. I am indepted to him for helping me with my thesis.

I also want to thank Dr. Supratik Bhattacharyya and Dr. Timothy Roscoe, who were my intern advisors at Sprint Labs. During my two internships at Sprint Labs, I acquired invaluable skills and had the pleasure to work on two very exciting summer projects.

A big thank you is due to Dr. Pablo Rodriguez who had been my intern advisor at Microsoft Research, and has been a close collaborator and very good friend ever since. Pablo had a significant influence both on my carrier, by helping me re-discover the joy of working on content distribution networks, and on my personal life, by convincing me to move back to Europe. Pablo, thanks a lot for your help!

My PhD benefited tremendously by discussions and collaborations I had during my graduate years with various faculty and fellow students both at Georgia Tech and in other universities. I want to start by thanking Mr. Dan Colestock, Dr. Stelios Kavadias, Dr. Pete Manolios, Dr. Vijay

Vazirani, Dr. Eric Vigoda, and the rest of my professors at Georgia Tech for helpful discussions and guidance. I want to thank my fellow students in the Networking and Telecommunications Group for providing a great working environment; I have benefited significantly by numerous discussions with them and by their helpful suggestions and comments about my work. I also want to thank Dr. Amin Saberi and Dr. Vangelis Markakis for interesting discussions and collaborations.

I also want to thank my many friends in Atlanta for their companionship during the six years I have spent in graduate school. Without them, my life in Atlanta would have been very boring.

Finally, I want to thank my parents and my sister Ioanna for their unconditional love and support. A supporting family is the most valuable asset in life; I owe them everything.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Complex networks like the Internet, peer-to-peer systems, and emerging sensor and ad-hoc networks are large distributed decentralized communication systems arising repeatedly in todays technology. In such networks it is critical to characterize network performance as the size of the network scales. The focus of this work is to relate basic network performance metrics to structural characteristics of underlying network topologies, and to develop protocols that reinforce and exploit desired structural characteristics.

We have used the notions of conductance and spectral analysis to study the structural characteristics of complex communication networks. These graph-theoretic notions are directly related to the performance of basic communication tasks performed on the graph of the network, including routing congestion, searching, and crawling. They can be also used to quantify and measure the presence of groups of nodes that connect preferentially with each other, which is also referred as clustering. The emergence of groups of nodes can be attributed to various economical, geographical, business, and technological reasons, and their presence affects the performance of basic network communication tasks.

For the case of the Internet at the Autonomous System (AS) level we show that (a) routing and congestion depends on conductance and spectral gap, (b) the spectral gap is sufficiently large for both Internet data and random graph models, hence, the conductance is large and congestion scales gracefully, (c) the spectral filtering method can be used to identify clusters with semantic properties which, in turn, can be used to define extremal traffic patterns.

For the case of peer-to-peer networks we first address the issue of dynamic topology maintenance and propose low overhead heuristics that result in topologies with good conductance.

The main objective of peer-to-peer networks is efficient content searching. We examine two search techniques, flooding and random walks, and relate their behavior to conductance and spectral gap. We show that searching by random walks gives better performance in terms of the distribution of the number of hits when the topology is clustered and when the topology is dynamic.

However, random walks are sequential processes and thus the expected termination time of searching via random walks is large. We study hybrid search schemes for unstructured peer-to-peer networks that combine random walks and flooding to achieve running times comparable to flooding and searching performance similar to random walks. We also propose a general searching scheme of which flooding and random walks are special instances and show how to use locally maintained network information to improve the performance of searching.

Our hybrid schemes for searching in peer-to-peer networks rely on local information, i.e. information that a node collects by monitoring the traffic going though its links. We wish to extend our understanding on how local metrics can increase the performance of basic communication tasks. We are specifically interested in the case of searching in peer-to- peer networks and in the case of topology construction and information aggregation in sensor and other ad-hoc networks.

CHAPTER I

INTRODUCTION

The success of the Internet has changed significantly the way we perceive, design, analyze, and reason about computer communication networks. The Internet has grown in two decades from a few hundred nodes to many thousand networks (Autonomous Systems) and millions of nodes and users. Moreover, this fast growth took place without a centralized entity that controls the evolution of the network and, despite pessimistic predictions [112, 134], the Internet performs extremely well [111, 134]. Networks that have managed to scale from few nodes to thousand and millions of nodes without centralized control and with seemingly unstructured organization and characteristics are often referred as complex networks.

The Internet is not the only complex communication network. Typical unstructured peer-to-peer networks, like Gnutella, eDonkey, and FastTrack, content distribution networks, like BitTorrent, and emerging sensor and ad-hoc networks share similar characteristics with the Internet. They scale from few to thousand and millions of nodes with no or limited centralized coordination, and their organization, including the topology, is seemingly unstructured.

What are the reasons that allowed these complex communication networks to scale? Are there any common characteristics that are shared among the different communication networks and which allowed them to scale efficiently without centralized coordination? If there are good primitives shared by all complex networks, then we should strive to exploit them further and design communication algorithms, protocols, and new networks that reinforce them.

Previous research has identified unique characteristics, namely degree distributions with heavily skewed statistics, of the topologies of many communication networks ([33, 52, 108]) and suggested the topology has a great impact on the performance of the network ([99, 135, 150]). The related question of how to construct topologies that realistically model the Internet for protocol evaluation and testing purposes has also attracted the interest of the research community [27, 30, 73, 159, 161]. It is interesting to note that the primitive of hierarchical organization, which is believed to support

graceful scaling of routing protocols, has been part of standard network models [30, 160, 161] and has also been observed in typical network topologies [148]. Note, however, that hierarchy is not a fundamental primitive, but comes as a result of the skewed degree distributions observed in typical Internet graphs, which, in turn, may result from simpler primitives like preferential connectivity ([15]) and local optimizations [9, 51]. Moreover, the skewed degree distributions imply the existence of a network core which carries a substantial fraction of the traffic and may contribute to congestion. How does congestion scales in the core of the network?

Another fundamental characteristic that has been observed in typical network topologies is clustering. In typical networks, groups of nodes connect preferentially with each other and have sparse connectivity to the rest of the network. (Connections between nodes belonging to the same cluster are referred as "intra-cluster" and the rest of the connections as inter-cluster.) Clusters may arise for various reasons: geographical constraints, common interests that result in increased intra-cluster traffic, protocol engineering Various definitions have been proposed to measure the clustering properties of the typical networks, like the clustering coefficient [27] and the assortativity metric [120]. However, previous definitions focus on measuring local clustering characteristics. Moreover, it is not known how such clustering definitions relate to network performance.

In this work we have proposed to use the spectral graph theory to study the properties of network topology. Spectral graph theory is the study of structural properties of graphs using algebraic techniques [37, 46, 47]. In a nutshell, spectral graph theory uses the connectivity matrix of a graph, which is a $0/1$ matrix with entry 1 in position (i, j) if there is an edge in the graph from i to j, or adaptations of it and computes the eigenvalues and the corresponding eigenvectors of the input matrix. These eigenvalues and eigenvectors relate to many important properties of the input graph, such as the diameter, the expansion, and others. In particular the stochastic normalization of the adjacency matrix, a matrix which has the entry $1/d_i$ in position (i, j) if there is an edge from i to j and where d_i is the degree of node i, is of particular importance, since its largest eigenvalues are related to the convergence properties of a random walk on that graph and they also relate to the capability of the network to route demand efficiently [61, 61, 83, 95, 98, 144].

The study of the eigenvalues and eigenvectors of matrices corresponding to graphs has been extensively used implicitly or explicitly in data mining. The fundamental problem of data mining is

the discovery of important information in large quantities of data. Similarly, current communication networks are very large structures and, more importantly, business and geographic relationships between networks and various design decisions are reflected in the topology of the network. Spectral analysis helps us identify and isolate properties of the topology. In Chapter 4 we have used spectral analysis to identify clusters of Autonomous Systems (AS) in the Internet topology with natural geographic and business relationships, and to define extremal traffic patterns [66].

In Chapter 3we study analytically and experimentally the effect of the existence of skewed degree distributions and of clusters of nodes on the congestion of the network [62]. We find both analytically and experimentally that the congestion of the core of the network scales well.

From an algorithmic point of view we want to design algorithms that take into consideration the unique properties of typical network topologies. Moreover, we are also interested in creating topologies that have good properties and, hence, for which we can design efficient protocols. Peer-to-peer systems and emerging sensor and ad-hoc networks will benefit from primitives that result in topologies with provable good connectivity and from efficient searching and information aggregation protocols. Spectral graph theory can be used to identify construction primitives that result in good topologies. It is known that the algorithmic primitive of taking trajectories of random walks on graphs with good connectivity has good statistical properties. In Chapter 5 we adapt this methodology to design and analyze searching and construction via random walks in peer-to-peer networks [63]. In Chapter 6 we propose hybrid searching schemes for peer-to-peer networks [64].

Related work is presented in Chapter 2. We summarize the contributions of this work in Chapter 7.

CHAPTER II

RELATED WORK

The focus of this work has been the study of communication networks, in particular the Internet and peer-to-peer networks, using techniques borrowed from graph theory. Hence, we have made extensive use of concepts and ideas borrowed from a variety of research areas. In the following, we describe the previous work that helped us to shape and answer our research questions and, also, comment on complementary to our research approach work.

In Section 2.1 we describe work related to understanding the properties of the Internet and, in particular, its topology, and, also, we describe efforts to model Internet's topology. In Section 2.2 we describe work related to the study and modeling of peer-to-peer systems. In Section 2.3 we describe work related to the algebraic study of topological properties of graphs, and to the study of other graph theoretic problems that influenced our work.

2.1 Internet topologies: Characteristics and Models

The study of the characteristics of the topology of the Internet at the Autonomous system level has been the focus of extensive research in the last years ([27, 33, 52, 59, 73, 91, 108, 135, 148, 150, 155]). In particular, in a pioneering work, Faloutsos et al. identified the existence of highly skewed distributions (power-laws) in various metrics of the Internet topology, such as the degrees, the rank, and others [52]. Research that followed identified more power-laws and other interesting properties of the topology, such as hierarchy ([33,59,108,148]). This line of work showed that the graph of the Internet is highly irregular and that there is a small subset of very important for global connectivity nodes (the core of the network). The existence of a core that may become a bottleneck that prohibits the further growth of the Internet motivated our study of the scaling properties of the Internet.

Another very important property of Internet topologies that influenced this work is clustering. The existence of clusters either directly or indirectly, through the use of various metrics of clustering such as the clustering coefficient, is well known [27, 120]. Previous work however mainly used

metrics that capture local clustering properties. However, it is not known how local metrics relate to network performance (even though, they can have other uses, including topology characterization and comparison).

The study of the structural characteristics of the the Internet graph is of interest if the topology affects the performance of basic communication tasks. Radoslavov et al. show how the topology affects basic multicast protocols [135]. Tangmunarunkit et al. ([150]) compares the Internet topology to synthetic topologies with respect to many metrics, including the resilience to link removals and the link value which is similar to our link load metric ([62]). In a different context, Li and Prasant shows how the topology of overlay networks affects their performance [99].

The properties of the topology of the Internet are very important for generating realistic topologies for protocol evaluation and testing. Many approaches have been proposed in order to generate topologies resembling with respect to various metrics the topology of the Internet both at the router level and at the AS level [5, 27, 30, 65, 73, 108, 109, 156, 161]. Our work defines extra network characteristics that are of interest to protocol evaluation and testing.

Akella et al. study the routing problem in graphs grown with preferential connectivity and argue that the scaling properties are bad (scaling $\Omega(n^{1+1/\alpha})$ compared to our result of $O(n \log^2 n)$) [8]. In their proof they assume routing through nodes with high degrees, when alternative paths exist, and, hence, they bias traffic toward the core of the network, when better paths do exist.

2.2 Peer-to-peer systems

Decentralized unstructured peer-to-peer systems, like Gnutella [67, 76, 122], FastTrack [78], and eDonkey [50], have become extremely popular for file sharing applications [77]. Despite early scaling problems [140], current peer-to-peer networks serve thousands and millions of users, who mainly use the network for content searching. Naturally, efficient searching has attracted the interest of the research community. The existence of a large and dynamic user population raised also questions of how to maintain the topology of the network in the presence of churn, i.e. user arrivals and departures.

Lv et al. studied the performance of searching in peer-to-peer networks and argued that random probing is a realistic model [105]. They have experimented with various topologies (power-law

random graphs, random graphs, real data) and observed that flooding does not perform well in power-law graphs and in measured Gnutella topologies. Moreover, [105] proposed novel searching mechanisms for peer-to-peer networks, including the use of random walks. Compared to this work, our work in Chapters 5 and 6 (and also in [63, 64]) explains in more detail the dependence of the performance of searching via both random walks and floodings on the underlying topology. Moreover, we propose adaptations of random walks and floodings with provable good performance, and we also experiment with much larger datasets. In subsequent to [105] work, [40,41] showed how to improve the performance of searching via controlling content replication and by using additional semantic information respectively.

The performance of content searching using random walks and, in addition, the use of 1-step replication to increase searching performance has been also addressed in [2, 3, 32, 106].

Searching via random walks is preferable to searching via flooding, in terms of the statistical properties of the number of hits, when the underlying topology has clusters. Clustering is a characteristic of peer-to-peer topologies as observed in [74, 139]. Observe that, in our work, we consider clustering in the network topology; clustering may also exist in the interests of the users [55, 71, 81, 93].

A variety of other techniques have been proposed to improve the performance of searching [32,40,143,149,152]. A different approach for solving the searching problem is the use of structured peer-to-peer systems [88, 137, 141, 147].

The issue of topology construction in peer-to-peer networks has recently received significant attention. The objective of all proposals is to create topologies with good connectivity, and, more importantly, topologies that enable efficient content searching. One of the first approaches was the use of structured peer-to-peer networks [88, 137, 141, 147]. Another approach is to construct topologies that explicitly create groups of nodes with natural interest or network proximity [81, 103, 110] Compared to these proposals, our main objective is to construct a topology with provable good performance; additional interest or proximity semantics can be build on top of our schemes.

Our approach for constructing peer-to-peer topologies is more related to the distributed construction of Hamilton cycles proposed in [92]. In both cases the objective is to construct an expander

graph. Our topology construction schemes are more efficient in terms of number of messages required per node arrival, and use simpler approaches that can be implemented in current networks. On the other hand, our approach is weakly decentralized. Pandarungan et al. proposes a server-based scheme for constructing topologies with good connectivity characteristics [125].

2.3 Spectral methods and approximation algorithms

Spectral graph theory has been studied in a long series of books and papers [37, 46, 47, 132, 133]. More importantly for our work is the relation of the eigenvalues of the graph to conductance, expansion, and, hence, global connectivity, [12, 75, 144]. The conductance of the graph has been also related to the statistical properties of trajectory sampling via random walks [7, 24, 43–45, 61]. Gillman shows the dependence of the topology, in particular its second largest eigenvalues, to the probability that a random walk on the topology diverges from its stationary distribution [61]. Broder and Karlin show the relation of the cover time of a random walk on a topology to the eigenvalues of the Markov chain that describes the random walk [24].

The conductance of a graph has been also related to the performance of multi-commodity flow problems [83, 95, 98, 154]. In particular Leighton and Rao have shown that for the n node multi-commodity flow problem with uniform demands, the maximum achievable flow that can be routed is within an $O(\log n)$ factor of the upper bounds implied by the capacity and the topology of the network (min-cut) [95, 98].

Spectral clustering has been used in data mining applications [14, 70, 126, 136] and web ranking analysis [86, 124]. In both cases spectral clustering uses results from spectral graph partitioning to identify hidden information in the input graph. We have used similar ideas to identify hidden information in AS topologies, in particular clusters of nodes, and to assign importance (ranking) information to ASes.

CHAPTER III

SCALING OF CONTENT COMMUNICATION NETWORKS

3.1 Introduction

By the mid 90's, when the exponential growth of the Internet became apparent, several predicted an imminent Internet collapse, among other reasons, due to lack of capacity. Recall the memorable quote: "The current onslaught of new users will be more than the Internet infrastructure can support. More new capacity will be needed than the companies that build that infrastructure can deploy" [112, 134]. Not only did this not happen [111], but experimental work suggested that the Internet is performing well with respect to many statistics [134]. The question is whether it was good engineering, inherent properties of the Internet, or, just luck that allowed it to survive this rapid growth without serious performance deterioration. In this chapter we argue that the Internet's topology has inherent structural properties that support routing with near-optimal congestion. In particular, we argue that a general family of "Power-Law Random Graphs" (PLRG) [5, 20], of which the Internet topology is believed to be an instantiation [5, 52, 73, 150], has good "expansion" properties. Consequently, approximation algorithms for multicommodity flow imply that this family of graphs supports routing with near-optimal congestion [154, Chapter 21].

Performance is a term that means different things to different parties and different applications. The users of a communication network are concerned with round-trip delay, packet drop probability and throughput. Service providers are concerned with congestion and efficient use of network resources. For applications, such as the WWW, P2P and streaming video, performance metrics become even more context specific. However, there are simple graph-theoretic abstractions believed to capture properties that are essential in "every good network." For example, small diameter (a well defined graph theoretic property) is a desirable characteristic [22, 34], and more recently, the small world phenomenon (a less well defined property, but generally understood to mean small average distance and good clustering) has been identified as a desirable characteristic in searching the WWW and P2P networks [85, 86].

Another graph theoretic abstraction that has been pervasive in algorithms and complexity over the last thirty years is that of *expansion* and its generalizations, namely *cut sparsity* and *conductance* [37, 96, 118, 127, 144, 154]. In a sequence of celebrated results, which generalize the max-flow min-cut theory (e.g. see [95, 101, 154]), cut sparsity and conductance have been explicitly correlated with the performance of routing algorithms as follows. Let $G(V, E, W)$ be an undirected capacitated graph, and let c_e, $e \in E$, denote the capacities. Let $\{(s_1, t_1), \ldots, (s_k, t_k)\}$ be specified pairs of nodes (where each pair is distinct, but a node may be present in several pairs). A separate commodity i is defined for each (s_i, t_i) pair, and for each commodity i a nonnegative demand $\mathrm{dem}(i)$ is also specified. For some routing of this demand, let $l_e \cdot c_e$ denote the flow through link e. The *congestion* of link e is the quantity l_e. Thus the *maximum link congestion* according to this routing is $L = \max_{e \in E} l_e$. The objective is to find a routing that minimizes the maximum link congestion. For example, if we were given the network's topology and demand, L would be the amount of provisioning in the thickest link.

When the routing respects capacity constraints, then l_e is the usual notion of utilization. We shall also consider routings that violate capacity constraints. What is then the meaning of l_e and L? If all demands were scaled by a factor $1/L$, then the capacities would be respected. (The quantity $1/l_e$ is also referred to as "throughput.") Thus, in analyzing L we are making the assumption that $1/L$ is indicative of the worst performance over all links.

Consider a cut $(S, \bar{S})$. Let $\delta(S)$ denote the links that have one endpoint in S and another endpoint in $\bar{S}$. Let $c(S)$ denote the total capacity of the links in the cut: $c(S) = \sum_{e \in \delta(S)} c_e$. Let $\mathrm{dem}(S)$ denote the total demand separated by the cut: $\mathrm{dem}(S) = \sum_{i : |\{s_i, t_i\} \cap S| = 1} \mathrm{dem}(i)$. A natural lower bound on L follows by simple averaging principles. In particular, for every routing,

$$\sum_{e \in \delta(S)} l_e c_e \ \geq \ \mathrm{dem}(S) \ , \ \forall \, S \subset V \ .$$

Thus,

$$L \ \geq \ \max_{S \subset V} \frac{\sum_{e \in \delta(S)} l_e c_e}{\sum_{e \in \delta(S)} c_e} \geq \max_{S \subset V} \frac{\mathrm{dem}(S)}{c(S)} = \frac{1}{\min_{S \subset V} \frac{c(S)}{\mathrm{dem}(S)}}.$$

The theory of maximum multicommodity flow developed over the last decade [154, Chapter 21] suggests that there exists a routing with maximum link congestion within a factor $O(\log k)$ of the above lower bound. Moreover, there are known polynomial time algorithms for finding such a

routing:

$$\frac{O(\log k)}{\min_{S\subset V}\frac{c(S)}{\mathrm{dem}(S)}} \geq L \geq \frac{1}{\min_{S\subset V}\frac{c(S)}{\mathrm{dem}(S)}}\ . \tag{1}$$

The *cut sparsity* associated with a specific demand is the crucial ratio

$$\min_{S\subset V}\frac{c(S)}{\mathrm{dem}(S)}\ . \tag{2}$$

Let $(d_1, d_2, \ldots, d_n)$ be the degrees of the graph. Let $D = \sum_{i=1}^{n} d_i = O(n)$ representing the fact that graph is sparse. Consider demand $O(d_u d_v)$ between all pairs of nodes u and v. This includes one unit of demand between all n^2 pairs of nodes as a special case. Define the volume of a set of vertices $S \subset V$ as $\mathrm{vol}(S) = \sum_{u\in S} d_u$. Then (1) implies (details in Theorem 1

$$\frac{O(n\log n)}{\Phi} \ \geq\ L\ \geq\ \frac{O(n)}{\Phi} \tag{3}$$

where Φ is the *conductance* and is defined as:

$$\Phi = \min_{S\subset V, \mathrm{vol}(S)\leq D/2}\frac{c(S)}{\sum_{u\in S} d_u} = \min_{S\subset V, \mathrm{vol}(S)\leq D/2}\frac{\sum_{e\in\delta(S)} c_e}{\sum_{u\in S} d_u}. \tag{4}$$

Expander graphs are families of regular graphs (all nodes have the same degree) with linear number of unit capacity links ($|E|$ can be as low as $3|V|$) for which the so-called *expansion factor* is constant:

$$\min_{S\subset V, |S|\leq |V|/2}\frac{\delta(S)}{|S|}\ =\ \Omega(1)\ .$$

For such graphs, and for one unit of demand between all n^2 pairs of nodes, (3) and (4) imply that there exists a polynomial time routing algorithm with maximum link congestion $O(n\log n)$. Alternatively, if all links have capacity $O(n\log n)$ then the demand can be routed with maximum link congestion bounded away from 1. This is optimal since, by simple counting considerations, most of the n^2 pairs have hop distance $\Omega(\log n)$. In a rather strong sense, expander graphs enable excellent resource allocation, since with a linear number of links they support routing with optimal congestion. Note that in arbitrary graphs routing one unit of traffic between every pair of nodes may result in congestion as bad as $\Omega(n^2)$. For example, in a complete binary tree each link incident to the root needs to carry flow $(n/2)^2$. In a tree grown with preferential connectivity there are links incident to the highest degree node that need to carry flow $\Omega(n^{\frac{3}{2}})$.

Admittedly, the known polynomial time algorithms that achieve provably optimal performance (via LP-duality and metric embeddings) are complex and involve non-integral flows [154, Chapter 21]. However, there are complementary results suggesting that near-optimal congestion (up to $\text{poly} \log n$ factors) can also be achieved with integral short paths and decentralized, on-line algorithms (e.g. see [57, 87].) Therefore, a constant expansion factor is thought of as an "excellent promise" for routing. Random regular graphs are long known to possess constant expansion [118, Chapter 5.6]. These, together with explicit constructions [104] have found many applications in networks: [129–131] for non-blocking networks, [96] for parallel architectures, [97] for circuit switching, [92] for peer-to-peer networks, to list just a handful that span three decades. All these applications involve expanders as carefully constructed mathematical and engineering artifacts.

The networking paradigm is shifting. Today's open, distributed and dynamic networks are no longer artifacts that we construct, but phenomena that we study. One of the first observed striking differences is in the distribution of the degrees of the underlying network topologies: The degrees of the WWW [15, 25], the Internet at the level of Autonomous Systems [52] and at the router level [150], and several other examples [150], all follow heavy-tailed statistics, often expressed as *power-laws*: The frequency of nodes with degree d is proportional to $d^{-\zeta}$, for some constant ζ typically between 2 and 3. At the same time, these remain sparse, linear-size networks (the average degree is around 4 for Internet topologies and around 8 for the WWW.) The main result of this chapter is:

> *Power Law Random Graphs (PLRG) can support routing of $O(d_u d_v)$ units of flow between each pair of vertices u and v with degrees d_u and d_v respectively, with congestion $O\left(n \log^2 n\right)$. This includes unit demand between all pairs of nodes as a special case.*

This is only a $\log n$ factor off from the congestion achieved by linear size regular graphs with constant expansion. Thus our result can be understood as follows. The skewed degree distributions of PLRGs result in a hierarchical organization, with nodes of (typically) high degree forming the "core" of the network [66, 148, 150]. This is reminiscent of a tree-like structure. Intuitively, we expect that links in the core carry more flow. Our result suggests that the bound $O(n \log^2 n)$ by which the flow scales in the core of PLRGs is closer to the bound $O(n \log n)$ of a robust flat structure, such

as an expander, rather than the bound $\Omega(n^{1+\epsilon})$ of a tree.

What is the implication of our result for real networks, like the Internet, that are believed to be instantiations of PLRGs? Even though our model is very simple to capture the full complexity of real systems, we believe that it carries a positive message. We view the moderate rate at which the congestion grows and the established strong conductance as an indication that, despite its decentralized uncoordinated dynamic growth, the Internet preserves good resource allocation and load balancing properties and prevents extreme fragilities and monopolies [49, 151, 158]. Admittedly, this is a subjective statement, but we believe that it is a firm starting point.

In summary, we have analyzed routing on PLRG's under the assumptions:

(a) All links have the same capacity.

(b) Demand $O(d_u d_v)$ between all pairs of nodes (unit uniform demand is a special case.)

(c) The objective is to minimize maximum link congestion.

(d) Flows can be fractional and involve paths of arbitrary length.

We made assumptions (a) and (b) due to lack of publicly available information about link capacities and demand patterns. For practical purposes and for the worst case analysis considered here, we believe that assumption (a) is not particularly restrictive. See the definition of conductance (4). Intuitively, we expect that unbalance in link capacities will favor links belonging to cuts for which $\frac{|\delta(S)|}{\sum_{u \in S} d_u}$ is small. That only helps conductance Φ.

We believe that assumption (b) is quite restrictive. In particular, it does not capture popular sites that may have low degree, but high demand. However, the notion of cut sparsity can be defined for arbitrary capacities and demands and the general methodology of (1) and (2) carries over. Thus the methodology of our work provides a starting point to study more general demand patterns, once such patterns become better characterized.

We believe that assumption (c) is not particularly restrictive, though this is also subjective. Assumption (c) essentially imposes worst case analysis. There are many other performance metrics in computer science theory and networking, but worst case analysis is a reasonable place to start.

Assumptions (c) and (d) imply that the objective is to minimize the maximum link congestion. In many networks of practical interest the objectives can be different. For example many protocols

minimize the hop count for every source-destination pair. We have experimentally measured the congestion in power-law random graphs under shortest-hop integral routing (in a set-up reminiscent of the Internet at the level of Autonomous Systems.) Our measurements indicate that the congestion still increases like $O(n\text{poly}\log n)$.

A key technical ingredient in our proofs is to show that the core of the PLRG has strong conductance properties. These have further implications, most notably on the spectral gap of the stochastic normalization of the adjacency matrix of the core of the graph. Such spectral gaps have found in the past many algorithmic applications [37, 118, 144, 154]. For example, they imply reliability and fast cover times and hitting times —the latter are related to crawling and searching. Thus, we believe that our bounds on conductance and spectral gap will be a useful tool in establishing further properties for Internet performance. (Similar bounds have been subsequently obtained for the model of growth with preferential attachment [114]). In passing, we also note that the rather sharp bounds obtained in Corollary 1 are further evidence that the spectrum of the stochastic normalization of the adjacency matrix is an important metric for Internet topologies (see also [52, 66, 113]).

The rest of this chapter is organized as follows. In Section 3.2 we discuss random graph models for graphs with skewed degrees, including Internet topologies, and formalize the random model that is suitable for our study.

In Section 3.3 we give a theoretical argument based on conductance and along the lines of (3) and (4) that shows that maximum link congestion is $O(n\log^2 n)$. This section contains the conductance and eigenvalue separation proofs. The analytical argument applies to random graphs under certain model restrictions. We view these restrictions as mild, but certainly, in a strict sense, they do not include the whole class of power-law random graphs. More importantly, by invoking approximation algorithms for multicommodity flows, they involve non-integral flows and centralized routing along paths that are not necessarily short.

In Section 3.4 we validate the $O(n\text{poly}\log n)$ congestion bound experimentally, for integral routing along shortest paths using as graphs real and synthetic Internet topologies. We further compare the congestion of Internet and Internet-like topologies to trees and 3-regular expanders. These are worst-case and best-case sparse random graphs with respect to congestion. The congestion in real and synthetic Internet topologies appears to scale similarly to that of expanders than trees.

3.2 Structural Models for Graphs with Skewed Degree Sequences

Random graph models producing graphs with skewed degree sequences fall into two general categories: evolutionary and structural. *Evolutionary* models identify growth primitives giving rise to skewed degree distributions. These primitives can be microscopic, such as multi-objective optimization [9, 51], and macroscopic, such as statistical preferential connectivity [15, 42, 89, 108]. The advantage of microscopic evolutionary models is that they may capture additional network semantics. Their disadvantage is that they are hard to simulate and notoriously hard to analyze (e.g. see [51].) This is due to the detailed optimization problems solved in each step, and the dependencies between steps; dependencies pose the biggest hurdle in probabilistic arguments. The advantage of macroscopic evolutionary models is that they are easy to simulate. Their disadvantage is that they are also quite difficult to analyze, due, again, to the dependencies between steps (e.g. see [19].) *Structural* models start with a given skewed degree distribution, perhaps a power-law predicting the degrees of a real network [1, 73], and interpolate a graph that matches the degree sequence (exactly or approximately) and satisfies certain other randomness properties [6, 65, 73, 150]. A big advantage of such structural models is their amenability to analytical treatment. By taking the degree sequence as a granted, most of the dependencies arising in the analysis of evolutionary models can be removed [5, 38, 113]. This has been also noted by mathematicians who have known several robustness properties in structural random graph models for some time [20, 116, 117], though the term used there is *configurational*. In addition, structural models have been found good fits for Internet topologies [150]. Therefore, we will use a structural model similar to the one in [5].

Let $\vec{d} = (d_1, d_2, \ldots, d_n)$ be a sequence of integers. The structural or configurational method generates a random graph as follows. First consider $D = \sum_{i=1}^{n} d_i$ mini-vertices; think of mini-vertices as lying in n *clusters* of size d_i, $1 \leq i \leq n$. Then construct a random perfect matching among the mini-vertices and generate a graph on the n original vertices as suggested by this perfect matching in the natural way: two vertices are connected with an edge if and only if at least one edge of the random perfect matching was connecting mini-vertices of their corresponding clusters [6, 20]. This is an uncapacited graph. Alternatively, we may generate a capacitated graph by assigning capacity c_e between vertices u and v proportional to the number of edges between the clusters of d_u

and d_v mini-vertices corresponding to u and v. Note that this is a general random graph model. It makes no assumptions on the starting sequence of integers.

A power-law random graph (PLRG) is an uncapacitated graph generated according to the structural method for a degree sequence obtained by sampling from a power-law distribution. In [6] it was shown mathematically that a PLRG consists of more than one connected components, almost surely, though it has a giant connected component, almost surely. Necessary and sufficient conditions under which a general degree sequence results in a connected graph in the structural model were obtained in [116, 117]. From the technical perspective, notice that it might be hard to argue about expansion on a random graph model that is not even connected. The intuition is that, what causes small isolated connected components in the entire graph, may cause small sets with bad expansion inside the giant component.

In [150] it was argued experimentally that the giant component of a PLRG matches several characteristics of real complex networks, and hence is a good candidate for generating synthetic Internet topologies. However, one notable discrepancy between PLRGs and topologies of real communications networks is that in real networks nodes of very small degree (customers) are much more likely to be connected to nodes of large degree (providers). This has been measured in [33, 120] and has been formalized in [120].

We will use a technical modification of PLRG that ensures connectivity, almost surely, and always connects nodes of degree 1 and 2 to nodes of degree greater than 3. In practice, we will construct a modified PLRG as follows. For a degree sequence $\vec{d} = (d_1, d_2, \ldots, d_n)$, we first consider a connected graph that satisfies the degree sequence exactly and is reminiscent of some Internet topology. For example, we may consider the graph generated by Inet [73], any connected graph that satisfies the degree sequence and some further randomness criterion [18, 150], or a Markov chain Monte Carlo simulation approach [65]. We perform iterated pruning of vertices of degrees 1 and 2, until we are left with a graph whose smallest degree is 3. (The significance of "3" is that this is the smallest constant for which random graphs are connected expanders; for example, random 3-regular graphs are almost surely connected expanders, while random 2-regular graphs are almost surely disjoint cycles.) Let $\vec{\delta} = (\delta_1, \delta_2, \ldots, \delta_n)$ be the degree sequence of the graph after the pruning. We consider a PLRG generated for the non-zero degree vertices of $\vec{\delta}$; we call this

PLRG a *core*. Note that cores can be parameterized according to the smallest degree vertex that they contain; this natural notion has been repeatedly observed whenever iterated pruning has been considered (e.g. see [26].) Finally, we attach the pruned vertices with their original degrees in a preferential fashion. This can be done as follows. For each vertex u in the core consider $d_u - \delta_u$ mini-vertices. Let U be the set of mini-vertices arising from vertices in the core. For each pruned vertex v consider d_v mini-vertices. Let V be the set of mini-vertices arising from pruned vertices. We may now construct a random maximum matching between U and V and connect u to v if and only if some mini-vertex arising from u was connected to some mini-vertex arising from v.

Ideally, the modified PLRG model described above can become completely formal, once the degrees $\vec{\delta}$ of pruned Internet topologies are characterized. We have measured these degrees in a method similar to [52] and have found them to also obey well characterizable heavy tailed statistics (this is not surprising, neither intuitively nor analytically.) In this write-up we refrain from further descriptions of these tedious but straightforward measurements, in particular, because they are not necessary for the analytic argument. Indeed, the analytic argument holds for any sequence of integers $\vec{d} = (d_1, d_2, \ldots, d_n)$ with

$$D = \sum_{i=1}^{n} d_i = O(n) \text{ and } \max\{d_i,\ 1 \leq i \leq n\} = O(n^{\frac{1}{2}}) \tag{5}$$

and any sequence $\vec{\delta} = (\delta_1, \delta_2, \ldots, \delta_n)$ with

$$\begin{array}{rcll} \delta_i & \leq & d_i & \forall i \\ \sum_{i:\delta_i=0} d_i & \leq & \sum_{i:\delta_i \neq 0} d_i - \delta_i & \\ |\{\delta_i\ :\ \delta_i > 0\}| & = & \Omega(n) & \\ \delta_i & = & \Omega(d_i) & \forall i\ :\ \delta_i > 0 \end{array} \tag{6}$$

Conditions (6) are straightforward. They say that there are enough degrees in the core to absorb the non-core vertices, that the core is a constant fraction of the entire network and that the degree of each vertex inside the core is proportional to its degree in the entire network.

Conditions (5) say that the network is sparse and that the maximum degree is $O(n^{\frac{1}{2}})$. We justify the latter by pointing out that it is true in the evolutionary model of growth with preferential connectivity and the structural PLRG. For the evolutionary model it was proved in [20]. For the structural

model it can be verified as follows. Consider any degree sequence produced by n independent samples drawn from a power law distribution with $\Pr[d_i = d] \simeq d^{-\zeta}$, $2 < \zeta < 3$. The probability that any drawn sample has value greater than $\Omega(n^{\epsilon})$ is $O(n^{-\epsilon\zeta})$. Thus the probability that all drawn samples are smaller than $O(n^{\epsilon})$ is $O(n^{1-\epsilon\zeta}) = o(1)$, for $\epsilon > \frac{1}{\zeta}$. Since $2 < \zeta < 3$, $\epsilon = \frac{1}{2}$ suffices.

3.3 The Conductance Argument

Theorem 1. *Let $\vec{d}$ and $\vec{\delta}$ be sequences of integers satisfying (5) and (6). Suppose that all links have unit capacity. Then, there exists a way to route demand $O(d_u d_v)$ between all pairs of nodes u and v such that all links have flow $O(n \log^2 n)$. Moreover, the routing can be computed in polynomial time.*

Proof. Every node that does not belong to the core, that is where $\delta_u = 0$, will transfer its demand uniformly to the core vertices to which it is attached. Because of (6), this imposes demand $O(\delta_u \delta_v)$ between all pairs of vertices in the core.

To argue about the routing of the demand in the core we may start from (2). In particular, where V' is the set of vertices belonging to the core ($V' = \{u \in V \,:\, \delta_u > 0\}$), the cut sparsity in the core is:

$$\begin{aligned} \min_{S \subset V'} \frac{c(S)}{\mathrm{dem}(S)} &\geq \Omega\left(\min_{S \subset V'} \frac{c(S)}{\sum_{u \in S} \delta_u \sum_{v \in \bar{S}} \delta_v}\right) \\ &\geq \Omega\left(\min_{S \subset V', \mathrm{vol}(S) \leq \mathrm{vol}(\bar{S})} \frac{c(S)}{\Omega(n) \sum_{u \in S} d_u}\right) \\ &= \frac{\Phi}{\Omega(n)} \,. \end{aligned}$$

From Lemma 2, for the core, we have $\Phi = \Omega(1/\log n)$. Thus the cut sparsity in the core is $\Omega(1/n \log n)$. Now (1) implies that there is a routing of all the demands with maximum link flow $O(n \log^2 n)$. □

We proceed to establish conductance for the core. This is done in Lemma 2. The main technical step in in Lemma 1 that follows.

Lemma 1. [Main Lemma.] *Let $\vec{d} = d_1 \geq d_2 \geq \ldots \geq d_n$ be a sequence of integers with*

$$d_n \geq d_{\min} = 3 \quad \text{and} \quad D = \sum_{i=1}^{n} d_i = O(n) \,.$$

Let $G(V, E, W)$ be a graph with capacities generated according to the structural random graph model of Section 3.2. The conductance of $G(V, E, W)$ is

$$\Phi(G) = \min_{S \subset V, \mathrm{vol}(S) \leq D/2} \frac{c(S)}{\mathrm{vol}(S)} \;\geq\; \Omega(1) \;,$$

with probability $1 - o(1)$.

Proof. For a positive constant α, we say that a set of vertices S with $k = \mathrm{vol}(S) \leq D/2$ is Bad if

$$\frac{c(S)}{\mathrm{vol}(S)} < \alpha n \;.$$

We will show that there exists a positive constant α such that

$$\Pr[\exists \; \mathsf{Bad} S] \;\leq\; o(1) \;. \tag{7}$$

The left hand side of (7) is

$$\sum_{k=d_{\min}}^{D/2} \Pr[\exists \; \mathsf{Bad} S, \; \mathrm{vol}(S) = k] \;. \tag{8}$$

Let us fix k in the above range. There are at most $\binom{D/d_{\min}}{k/d_{\min}}$ sets of vertices in G that have volume k. This is because every such set arises from a set of mini-vertices such that the total number of minivertices is k and, for each cluster, either all or none of the mini-vertices of the cluster are included. Since the minimum cluster size is $d_{\min}$, the number of possibilities is maximized if all clusters were of size $d_{\min}$. We hence need to bound

$$\sum_{k=d_{\min}}^{D/2} \binom{D/d_{\min}}{k/d_{\min}} \Pr[\text{a fixed set } S, \mathrm{vol}(S) = k, \text{ is } \mathsf{Bad}] \;. \tag{9}$$

We analyze the probabilities that appear in the terms of the above summation. We may now assume that the set S is fixed. Let A denote the set of the k mini-vertices corresponding to S. Let $\bar{A}$ denote the set of the $(D - k)$ mini-vertices corresponding to $\bar{S}$. Let $B_A \subset A$ be the set of mini-vertices in A that were matched to mini-vertices in $\bar{A}$. Let $B_{\bar{A}} \subset \bar{A}$ be the set of mini-vertices in $\bar{A}$ that were matched to mini-vertices in A. In order for S to be Bad, the cardinality $|B_A| = |B_{\bar{A}}|$ is at most αk. For each cardinality in the range 0 to αk, there are at most $\binom{k}{\alpha k}$ ways to fix the mini-vertices in B_A and at most $\binom{D-k}{\alpha k}$ ways to fix the mini-vertices in $B_{\bar{A}}$. We may now assume that the sets B_A and $B_{\bar{A}}$ are fixed. We need to analyze the probability that the random perfect matching on the

mini-vertices matched all mini-vertices in $A \setminus B_A$ inside $A \setminus B_A$, all mini-vertices in $\bar{A} \setminus B_{\bar{A}}$ inside $\bar{A} \setminus B_{\bar{A}}$, and all vertices in $B_A \cup B_{\bar{A}}$ inside $B_A \cup B_{\bar{A}}$. The above probability can be expressed in terms of the total number of perfect matchings on n vertices. Let $f(n) = \frac{n!}{2^{n/2} \cdot (n/2)!}$ be the number of perfect matchings on n vertices. We may now write

$$\Pr[\text{a fixed set } S, \mathrm{vol}(S) = k, \text{ is } \mathsf{Bad}] \leq \alpha k \binom{k}{\alpha k} \binom{D-k}{\alpha k} \frac{f(2\alpha k)\ f(k-\alpha k)\ f(D-k-\alpha k)}{f(D)} \tag{10}$$

We proceed with calculations. We bound each term of (10) separately. We will repeatedly use the following bounds which follow from Stirling's approximation [20, p.4]. There are positive constants c_1 and c_2 such that,

$$c_1\, n^{n+1/2}\, e^{-n} \ < \ n! \ < \ c_2\, n^{n+1/2}\, e^{-n}. \tag{11}$$

Using $\binom{n}{m} = \frac{n!}{m!\,(n-m)!}$ and (11), for some constant c_3, we can bound

$$\binom{D/d_{\min}}{k/d_{\min}} < c_3 \left(\frac{D^D}{k^k\ (D-k)^{D-k}} \right)^{1/d_{\min}}. \tag{12}$$

Using the inequality $\binom{n}{m} \leq (n\,e/m)^m$, we bound

$$\binom{k}{\alpha k} \binom{D-k}{\alpha k} \leq \left(\frac{e}{\alpha}\right)^{2\alpha k} \left(\frac{D-k}{k}\right)^{\alpha k} \tag{13}$$

By substituting the values of f and using (11), for some constant c_4, we can bound

$$\begin{aligned}
&\alpha k\, \tfrac{f(\alpha k)\, f(k-\alpha k)\, f(D-k-\alpha k)}{f(D)} && = \\
&\alpha k\, \tfrac{(2\alpha k)!}{2^{\alpha k}(\alpha k)!}\, \tfrac{(k-\alpha k)!}{2^{k(1-\alpha)/2}(k(1-\alpha)/2)!} \cdot \tfrac{(D-(1+\alpha)k)!}{2^{\frac{D-(1+\alpha)k}{2}} \left(\frac{D-(1+\alpha)k}{2}\right)!}\, \tfrac{2^{D/2}(D/2)!}{D!} && < \\
&c_3\, \alpha k \tfrac{2^{2\alpha k} 2^{1/2} (\alpha k)^{2\alpha k}}{(\alpha k)^{\alpha k}}\, \tfrac{(k(1-\alpha))^{k(1-\alpha)}\, 2^{k(1-\alpha)/2} 2^{1/2}}{(k(1-\alpha))^{k(1-\alpha)/2}} \cdot \tfrac{(D-(1+\alpha)k)^{D-(1+\alpha)k}\, 2^{\frac{D-(1+\alpha)k}{2}}}{(D-(1+\alpha)k)^{\frac{D-(1+\alpha)k}{2}}}\, \tfrac{D^{D/2}}{D^D\, 2^{D/2}} && < \\
&c_4\, \alpha k\ (2\alpha)^{\alpha k}\ \tfrac{k^{k(1+\alpha)/2}\, (D-k)^{(D-k-\alpha k)/2}}{D^{D/2}} && .
\end{aligned} \tag{14}$$

Now combining (12), (13) and (14) we get that, for some constant c_5,

$$\binom{D/d_{\min}}{k/d_{\min}} \binom{k}{\alpha k} \binom{D-k}{\alpha k} \alpha k \, \frac{f(\alpha k)\, f(k-\alpha k)\, f(D-k-\alpha k)}{f(D)} < \tag{15}$$

$$c_5 \alpha k \frac{2^{\alpha k}\, e^{2\alpha k}}{\alpha^{\alpha k}} \left(\frac{k}{D}\right)^{k((1-\alpha)/2-1/d_{\min})}$$

Define

$$\beta = \frac{2e^2}{\alpha} \text{ and } \gamma = \frac{1-\alpha}{2} - \frac{1}{d_{\min}} \,. \tag{16}$$

Define

$$G\left(k\right) = c_5 \alpha k \, \beta^{\alpha k} \left(\frac{k}{D}\right)^{\gamma k} \,. \tag{17}$$

Using (10), (15), (16) and (17), we can bound the quantity in (9) by

$$\sum_{k=d_{\min}}^{D/2} G\left(k\right)$$

Note that a necessary condition for the sum to be bounded is that the terms $G(k)$ become vanishing, which, by (17), requires $\gamma > 0$, which, by (16) implies $d_{\min} > \frac{2}{1-\alpha} \Rightarrow d_{\min} \geq 3$ and $\alpha \leq \alpha_1(d_{\min}) = 1 - \frac{2}{d_{\min}}$.

The first derivative of $G(k)$ is

$$\frac{d\, G(k)}{d\, k} = \left(\frac{1}{k} + \alpha \ln \beta + \gamma \ln \frac{k}{D} + \gamma\right) G(k) \tag{18}$$

The second derivative of $G(k)$ is

$$\frac{d^2\, G(k)}{d\, k^2} = \left(-\frac{1}{k^2} + \frac{\gamma}{k} + \left(\frac{d\, G(k)}{d\, k}\right)^2\right) G(k) \tag{19}$$

The first derivative is negative for $k = 3d_{\min}$ and sufficiently large D. The second derivative is positive for $k \geq 3d_{\min}$ and $\alpha \leq \alpha_2(d_{\min}) = 1 - \frac{8}{3d_{\min}}$. Notice that $d_{\min} \geq 3$ guarantees that α is positive. Thus $G(k)$ attains its maximum either at $G(3d_{\min})$ or at $G(D/2)$. We wish to bound both these quantities by $o(1/D)$.

$$G(3d_{\min}) = c_6 \frac{1}{D^{3d_{\min}\gamma}} \leq o\left(\frac{1}{D}\right)$$

for constant c_6 and $3d_{\min}\gamma > 1$ which implies $\alpha \leq \alpha_2(d_{\min})$.

$$G(D/2) = c_7 \frac{D}{2} \left(\frac{\beta^\alpha}{2^\gamma}\right)^{D/2} \leq o\left(\frac{1}{D}\right)$$

for constant c_7 and

$$\beta^\alpha < 2^\gamma \Rightarrow \alpha \left(\frac{3}{2} + 2\log_2 e - \log_2 \alpha \right) < \frac{1}{2} - \frac{1}{d_{\min}} .$$

The left side of the inequality is a monotonically increasing function of α in the range $(0, 1]$. A sufficient condition for the inequality to hold gives the third condition for α: $\alpha < \alpha_3(d_{\min})$ ($\alpha < 0.0175$ for $d_{\min} = 3$ suffices).

For k in the range $d_{\min}$ to $3d_{\min}$, there are a constant numbers of terms of (17). It can be seen that each one of these terms is $o(1)$.

Thus:

$$\begin{aligned} \sum_{k=d_{\min}}^{D/2} G(k) &= \sum_{k=d_{\min}}^{3d_{\min}-1} G(k) + \sum_{k=3d_{\min}}^{D/2} G(k) \\ &\leq o(1) + D \cdot o(1/D) \\ &= o(1) \end{aligned}$$

This completes the proof of (7) by using (9), (15) and the definition of $G(k)$ (17). □

Lemma 2. *Let $\vec{d} = d_1 \geq d_2 \geq \ldots \geq d_n$ be a sequence of integers with*

$$d_1 = O(n^{\frac{1}{2}}), \quad d_n \geq 3 \quad \text{and} \quad D = \sum_{i=1}^{n} d_i = O(n) . \tag{20}$$

Let $G(V, E, W)$ be a graph with capacities generated according to the random graph model of Section 3.2. Let $G(V, E)$ be the corresponding uncapacitated random graph. The conductance of $G(V, E)$ is

$$\Phi(G) \geq \Omega(1/\log n) ,$$

with probability $1 - o(1)$.

Proof. In Lemma 1 we showed conductance $\Omega(1)$ for the capacitated $G(V, E, W)$. Therefore it suffices to show that no link will have capacity more than $O(\log n)$, almost surely. In turn, it suffices to bound the probability that the link between vertices u and v of degrees d_1 and d_2 respectively have capacity more than $O(\log n)$. We will bound this probability by $o(1/n^2)$. This probability is maximized when $d_1 = d_2 = \Theta(n^{\frac{1}{2}})$.

Let u_1, ..., u_{d_1} be the mini-vertices corresponding to u. Let Y_{u_i}, $1 \leq i \leq d_1$, be 1 if u_i is connected to a mini-vertex corresponding to the cluster of minivertices of v and 0 otherwise. We are interested in the capacity $Y = \sum_{i=1}^{d_1} Y_{u_i}$. This can be bounded by the sum X of $d_1 = \Theta(\sqrt{n})$ independent Bernoulli trials, each one with probability of success $\frac{d_1}{D-d_1} = \Theta(1/\sqrt{n})$. The expectation $\mu = E[X] = \Theta(1)$.

Using the standard tail equality (Chernoff bound) is [20, p.12]:

$$\Pr[X > \mu + t] \ < \ e^{-\frac{t^2}{2(\mu + t/3)}}$$

we get

$$\Pr[Y > \Omega(\log n)] = o(1/n^2) \ .$$

□

Corollary 1. *Let $G(V, E)$ be a random graph as in Lemma 2. Let A be the adjacency matrix of G. Consider a stochastic matrix P corresponding to a random walk G. The spectrum of P is in $[-1, 1]$, with the largest eigenvalue $\lambda_1 = 1$. Let λ_2 be the second largest eigenvalue. Then,*

$$1 - \Omega\left(\frac{1}{\log n}\right) \ < \ \lambda_2 \ < \ 1 - \Omega\left(\frac{1}{\log^2 n}\right) \ .$$

Proof. Follows from the known inequalities (e.g. see [144, p.53])

$$1 - 2\Phi \ < \ \lambda_2 \ < \ 1 - \frac{\Phi^2}{2}$$

and Lemma 1. □

3.4 Evaluation

3.4.1 Methodology

In the previous sections we proved congestion properties for routing involving non-integral flows over paths of arbitrary length. In this section, we strive to experimentally verify that the basic conclusions hold even under shortest-hop routing involving integral flows.

A canonical example of particular interest is Internet routing at the level of Autonomous Systems (AS). The routing protocol at the AS level is the Border Gateway Protocol (BGP) [138]. This

protocol filters all paths that do not satisfy certain policy constraints and among the remaining ones, it picks paths with minimum hop distance. The policy constraints can be arbitrarily complex, but at minimum they are set in such a way to prevent transit traffic [59]. Network owners are willing to route traffic that originates or terminates in their own network or in the networks of their (paying) customers. They do not allow traffic arriving from one of their (non-paying) peers to be forwarded to another peer, since this wastes network resources without generating revenues.

We model two routing schemes. The first one is *Shortest-Hop Routing*. This scheme routes along paths with minimum hop (AS) distance. In case of multiple minimum hop paths, we pick one of them at random. The second is *Shortest-Hop Routing with Policies,* which is an extension of the previous scheme without transit traffic. The first scheme can be applied to all topologies. The second scheme can be used only when information about the type of the links is available, which is the case for the real topologies and not for the synthetic ones.

As in the theoretical analysis, we have assumed that there is one unit of demand between any two ASes, which is routed over a single shortest paths. For each link we compute the number of paths going through that link (link congestion), and we examine the maximum number over all links (congestion). We also examine the profile of these values. We study how the maximum link congestion evolves as the size of the network increases and how the rate of increase compares to two baseline models. The first baseline model is the family of 3-regular expanders. Graphs of this type have congestion $O(n \log n)$. The second baseline model is the family of trees grown with preferential connectivity. These graphs have congestion $\Omega(n^{3/2})$. To see this, we use the fact that such trees have a node of degree $\Omega(n^{1/2})$ [22]. Let u be this node. Let v be a neighbor of u such that the subtree rooted at v has $\Omega(n^{1/2})$. Let K be this subtree. If $|K| < \frac{n}{2}$, then the complement of K has at least $n/2$ nodes. Thus, the link (u, v) carries flow $n^{1/2}\frac{n}{2}$. If $|K| \geq \frac{n}{2}$, then the link (u, v) must carry flow between K and the neighbors of u outside K. This is $\frac{n}{2}(n^{1/2} - 1)$. We find qualitatively that the congestion of the AS topology is closer to 3-regular expanders than trees grown with preferential connectivity.

In Section 3.4.2 we discuss the data used. In Section 3.4.3 we discuss the evolution of congestion with time. Further observations are in Section 3.4.4.

3.4.2 Data Used

We have used AS topology data from two sources. The first source is [4]. In addition to the topology, [4] classifies links as customer-provider, or peer-to-peer. This classification is important for policy routing. We have data from [4] for the years 2001 and 2002.

The second set of data is [121]. Though this set is far less complete, it has the advantage that it spans the time period of 1997 to 2001. [121] does not contain information about the relationships between the ASes. We have used the algorithm of [59] to infer AS relationships.

Using these data to derive conclusive results raises two issues. First, we have no way of knowing how complete the measurements are. Second, in order to study the evolution of the Internet from 1997 to 2002, we had to rely on data from two different sources which have different levels of accuracy [33]. Thus, the data are not directly comparable and we can observe minor irregularities in the results. Nevertheless, we believe that the trends we observe are correct.

To study the evolution of the Internet we need to experiment with topologies that are larger than the current Internet. We use the modified structural random graph model described in Section 3.2 to generate Internet-like topologies for large sizes with degree sequences obtained from Inet [73].

Note that there is an alternative representation of the Internet at the level of routers. The resulting graph contains much more detailed connectivity information. Even though the AS graph is a less detailed representation of the Internet than the router-level graph, it has three main advantages. First, it is smaller and thus amenable to processing. Second, detailed data for the AS topology are collected since 1997 [121]. No router level data that span that many years exist. Third, bottlenecks in the Internet are usually either at the access links, or at the connections between ASes (on the contrary, links inside the ASes are usually overprovisioned) [123].

The construction of trees with preferential connectivity and 3-regular expanders is straightforward. Our 3-regular expanders of size n are random graphs with n nodes, where each node has degree 3 (technically this is achieved by superpositioning three random perfect matchings.)

3.4.3 Evolution of Congestion with Time

First, we examine the evolution of the maximum link congestion over all links for the AS topology. The results for various snapshots of the Internet topology are given in Table 1. The number of

Table 1: Congestion of the AS topology

Year	Nodes	Links	Shortest Hop	Shortest Hop with Policies
1997	3055	5238	559653	1213810
1998	4341	7949	681067	2184841
1999	6209	12206	1051782	3914276
2000	9318	18272	1754897	8376841
2001	10915	23757	5523085	7533275
2002	13155	28317	9096654	11361893

Note: Maximum congestion over all links for different instances of the AS topology, assuming min-hop routing and min-hop routing with policies.

nodes increased by a factor of 4.3 between 1997 and 2002. The congestion increased by a factor of 13.36 and 9.36 for graphs with shortest hop routing without and with policies respectively. These results agree qualitatively with Section 3.3, which argues that congestion increases as a function of $n \cdot \text{poly} \log n$, where n is the number of nodes.

The transition from year 2000 to 2001 in Table 1 may come as an anomaly since the congestion for shortest hop routing with policies decreased. We believe that this is due to the fact that we used data collected from different sources.

To study the evolution of the congestion as the size of the network increases, we need to experiment with graphs that have a wider range in sizes. We have used the Inet generator to get an initial degree sequence $\vec{d}$. We used the Markov-Chain method to get a random graph for $\vec{d}$ [65]. We used the method of Section 3.2 to get a degree sequence $\vec{\delta}$ for the core. We used the modified PLRG model of Section 3.2 to get the final topology Figure 1 gives the maximum link congestion for graphs of size 3037, 5000, 7000, ..., 23000. The observation for this figure is the shape of the curve. The values of the congestion are bounded from above and below by $O\left(n \cdot \log^{2.6} n\right)$ and $O\left(n \cdot \log^2 n\right)$ respectively. These bounds were not designed to be tight, but to illustrate that congestion in Internet-like topologies grows as a function of $O\left(n \cdot \log^k n\right)$ for a small value of k. The observations remain the same if we use the output of Inet, or the unmodified PLRG.

How does the congestion of synthetic PLRG compares to that of 3-regular expanders and trees? We give this comparison in Figure 2. The modified PLRG model appears to behave much closer to the 3-regular expander than the tree. The evolution of congestion for 3-regular expanders and trees are given in Figures 3 and 4 respectively.

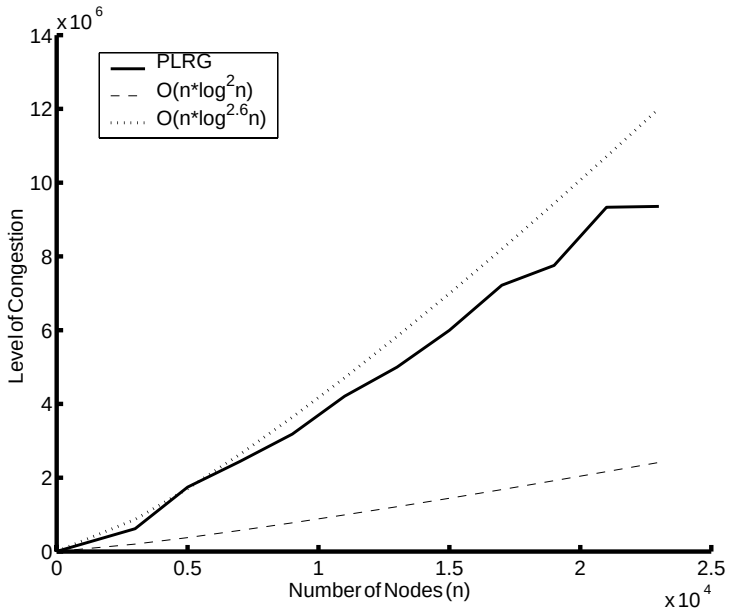


Figure 1: Congestion for PLRG topologies.

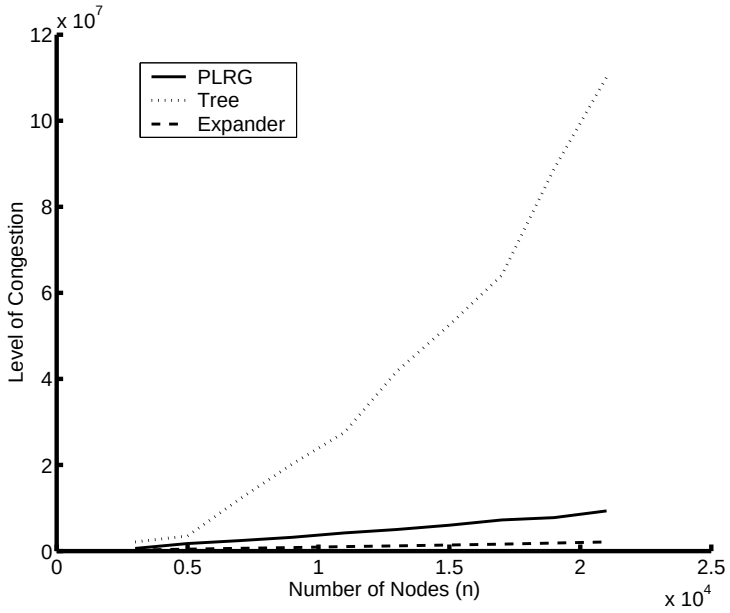


Figure 2: Comparison of congestion between synthetic PLRGs, 3-regular expanders and trees grown with preferential connectivity.

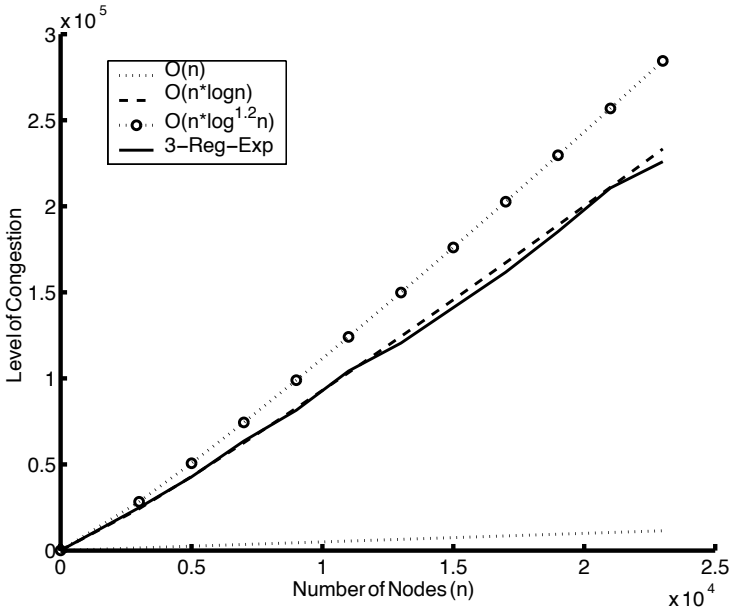


Figure 3: Congestion for 3-regular expanders.

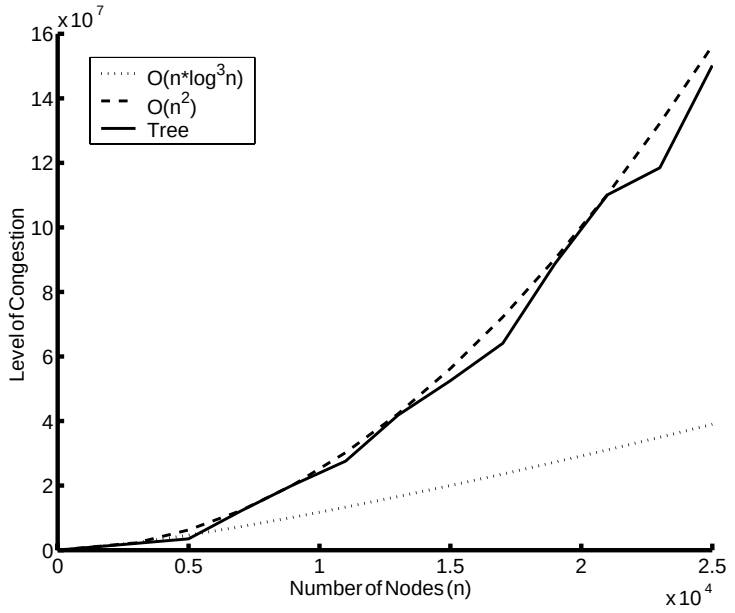


Figure 4: Congestion for tree grown with preferential connectivity.

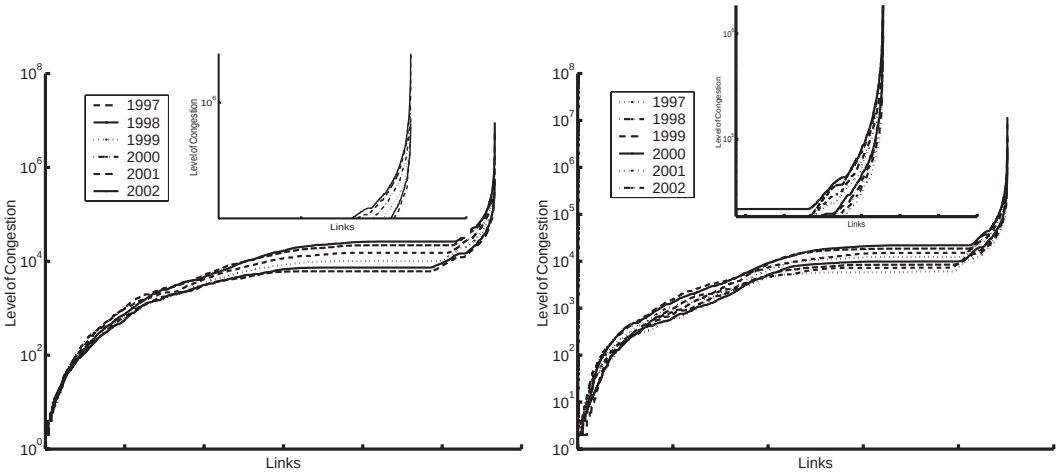


Figure 5: Congestion Fingerprint: Congestion of all links for various snapshots of the AS topology assuming min-hop routing (left) and min-hop routing with policies (right). Links are sorted from the least congested to the most. Observe the log scale in the vertical axis. We have stretched the curves horizontally to be comparable in size. This is important since different topologies have different number of edges.

3.4.4 Congestion Fingerprints

Next, we examine in more detail the characteristics of the link congestion for all links. Consider all the links of the graph sorted according to their congestion in increasing order. We call the resulting profile the *congestion fingerprint* of the graph.

The congestion fingerprint of the AS topology is drawn in Figure 5. The fingerprint is given for various snapshots of the Internet and for the routing with and without policies. Below, we discuss some further observations.

The most congested links have always one endpoint (and very often both) of high degree (see Table 2.) Nodes with high degrees correspond to big providers. Thus, congestion appears around the core.

The difference between the most congested links and the average congestion over all links is 2-3 orders of magnitude. The difference between the most congested links and the least congested ones is 7 orders of magnitude. Observe that the maximum possible difference is bounded above by n^2, which is less than 210^8 for graphs with $n < 14K$.

Around 10% of the links are heavily congested. These are the peaks of Figure 5. In this sense, our work can be understood as quantifying $O\left(n \cdot \mathrm{poly}\log n\right)$ as the growth rate of the "peak."

Table 2: Degrees of endpoints for the ten most congested links.

Congestion	Degree 1	Degree 2
9096654	2640	625
4223943	2640	1530
3704681	2640	586
3267646	2640	795
2742594	2640	159
2481002	2640	140
2188507	2640	837
1714812	586	1530
1682390	2640	330
1681173	2640	191

Results for a topology of 13155 nodes using integral shortest hop routing without policies.

Both fingerprints show the same trends. This means that the use of policy routing does not give extra information for our purposes. Thus, we believe that the analysis of non-policy routing in the synthetic topologies approximates well routing with policies for our purposes.

Using the degree sequences of real topologies, we generated synthetic topologies using the PLRG and the modified PLRG. The congestion fingerprints of the generated graphs are qualitatively similar to the real topologies and very different of that of expanders (see Figures 6 and 7.)

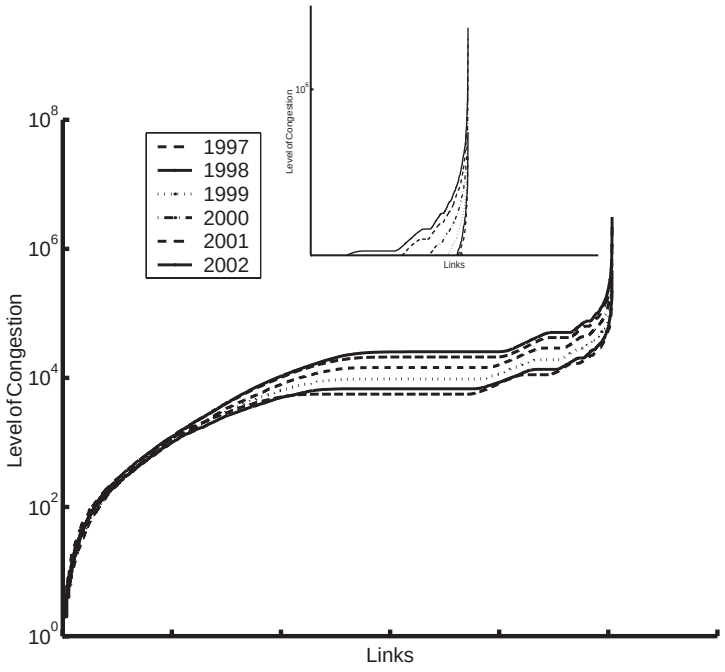


Figure 6: Congestion fingerprint for PLRG.

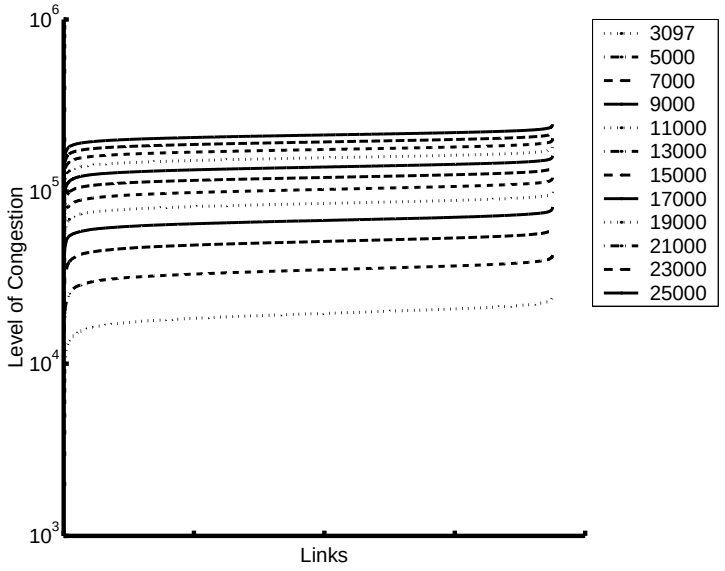


Figure 7: Congestion fingerprints of the 3-regular expander topologies (vertical axis in log scale.)

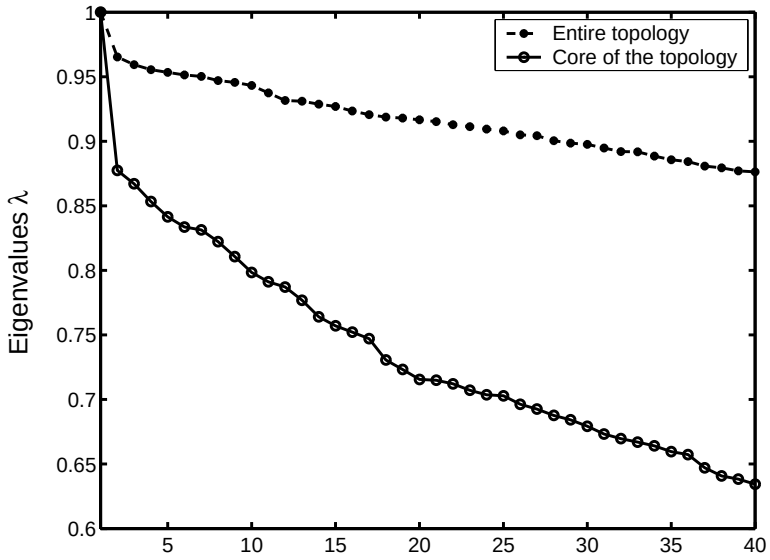


Figure 8: Eigenvalues of the Internet topology: The biggest eigenvalues of the stochastic matrix arising from the Internet at the AS level, for both the core and the entire topology [25]. Note the gap between the first and second eigenvalue in the core of the topology and see Corollary 3.4.

CHAPTER IV

SPECTRAL CLUSTERING OF INTERNET TOPOLOGIES

4.1 Introduction

Studying and modeling network topologies is necessary for protocol performance evaluation and simulation of a variety of network problems. Early modeling efforts focused around random graphs with relatively regular degree distributions [30, 156, 160, 161]. With the rapid growth of the network and the persistent effort of network measurement [29, 82, 121], real topology data started becoming available, in particular at the AS (Autonomous System) level. Using such data Faloutsos et al. first observed that the degree distribution of the AS level topology is actually consistently highly skewed [52]. Consequently, the research community has shown considerable interest in obtaining topology models that better resemble the real data [5, 27, 73, 108], as well as understanding the impact of such network topologies on the performance of network protocols [135, 150].

This new generation of synthetic Internet topology models is strongly driven by the observed skewed statistics of the degree sequence and its evolution, and by even further observations of more detailed graph theoretic characteristics of the network. Most notably, following the natural intuition that, for example, geography must be relevant in the real Internet topology, Bu and Towsley paid special attention to the "clustering" coefficient [27]; the observation of the significance of geography has been also made by Yook et al. [159] and by Lakhina et al. [91].

In this chapter we revisit the issue of clustering. As opposed to previous work that has focused on the clustering coefficient, our starting point is the method of *spectral filtering*. This method examines the large eigenvalues of matrices related to the adjacency matrix, and looks for clusters in the eigenvectors associated with these eigenvalues. Indeed, the first reference to the large eigenvalues of the adjacency matrix of the AS Internet topology is the "eigenvalue power-law" which was reported together with the "degree power-law" [52]. The connection between spectral filtering and graph connectivity, including clustering, has been extensively studied in discrete mathematics (e.g. see the books of Chung [37] and Sinclair [144] and the further references that they point to), and

has found very successful applications in information retrieval and data-mining where clusters represent groups of data with semantic proximity [14, 70, 86, 126, 136]. Practical experience suggests that spectral analysis might be better suited for data that lack regularity (thus it has been extensively used in computer science), while clustering coefficients are better suited for data that have stronger regularities (thus they have been extensively used by physicists who study lattices, crystals, etc.). Indeed, by definition, spectral filtering yields a large number of clusters, and it can be applied iteratively in subgraphs of a network. By contrast, it is not clear how to grow clusters around nodes with large clustering coefficient and this approach is not typical in information retrieval or data-mining[1].

Our contributions include:

- Adaptation of the spectral filtering method in the context of the AS Internet topology, by (a) performing inverse frequency normalization via stochastic matrices, (b) considering similarity transformations and (c) considering the entire topology as well as subgraphs of the topology. As a result, we get non-trivial groupings of ASes with clear semantic proximities, such as geography and business interests.

- The observation that spectral filtering gives clustering results only after performing suitable normalizations on the adjacency matrix of the AS topology. On the other hand, the eigenvectors related to the largest eigenvalues of the adjacency matrix, without any normalization, examined by Faloutsos et al. [52], do not express interesting clusters. This is an experimental validation of the result of Mihail and Papadimitriou [113]. Subsequently to our work, Gkantsidis et al. [62], Mihail et al. [114], and Chung et al. [35, 36] obtained analytical characterizations of the large eigenvalues of the stochastic normalization and normalized Laplacian for various power-law graph models and made explicit connections with performance metrics, such as congestion, throughput, crawling, and searching.

- The observation that the clustering properties (a)vary in the core and the edge of the network and across geographic areas, (b)persist over time, and (c)are not accurately matched by synthetic Internet topology generators, though the Power Law Random Graph (PLRG) model

[1]Though a related approach called "k-means" is quite common; but we do not expand further on it, since we do not use it in this work.

comes close [5].

- Study of the connection between the information retrieved by spectral filtering and link stress (link stress can be thought of as a first approach towards congestion). In particular, we argue that the eigenvectors associated with the largest eigenvalues can be suggestive of non-trivial intra-cluster traffic patterns that cause significant decrease in the link stress. The decrease is much more notable in the Internet than in any synthetic topology. On the other hand, if the traffic patterns become inter-cluster, then the link stress increases. This reasoning is in line with Fabrikant et al. [51] and Carlson et al. [31] which suggest that network characteristics should be studied in the context of the design problem that they are trying to solve.

- A method to define intra-cluster and inter-cluster "traffic" patterns. These are patterns that deviate from uniform treatment of all pairs of nodes, and may represent "good" and "bad" test cases for network performance.

- A detailed and efficient AS ranking method according to the first eigenvector of a suitably defined stochastic matrix, which has strong correlation with other known hierarchical assignments [148]. This approach is an adaptation of the pagerank used by Google [124]. An adaptation of the same method for ranking links between ASes, found that rankings are highly correlated with link stress under uniform traffic. A further adaptation of the method to obtain groups of ASes that correspond to seemingly highly stressed cuts.

The rest of this chapter is organized as follows. In Section 4.2 we cover necessary primitives from linear algebra and highlight the intuition behind the spectral filtering method. We also introduce normalizations and similarity transformations, and discuss their suitability and necessity for graphs with skewed statistics, like the Internet topology. In Section 4.3 we describe the spectral filtering results for the AS Internet topology, and give the qualitative nature of the information retrieved by the eigenvectors. In Section 4.4 we give an application of the information retrieved by the eigenvectors in terms of defining non-trivial traffic patterns that deviate from uniform traffic. In Section 4.5 we give a method of ranking ASes and links between ASes that is highly correlated with hierarchical assignments.

4.2 Spectral Analysis of Matrices arising from Graphs

In this Section we give a high level overview of the intuition and the primitives of spectral filtering. We discuss the basics of eigenvalues and eigenvectors of matrices, some useful transformations and normalizations, and why the eigenvectors corresponding to the large eigenvalues contain information relevant to clustering. This motivates the processing that we will perform to the eigenvectors of the AS Internet topology in Section 4.3. We also observe that spectral filtering does not give clustering results without performing suitable normalizations on the adjacency matrix of the AS topology (see also [113]).

4.2.1 The Spectral Filtering Method

Let $G(V, E)$, $|V|$=n, be an undirected graph and let A be its adjacency matrix: a_{ij}=1 if $(i, j) \in E$, a_{ij}=0 otherwise. Since G is undirected, A is symmetric $a_{ij} = a_{ji}$. In general, the (i, j)-th entry of a symmetric matrix can be thought of as a measure of the correlation between parameters i and j. Let $\vec{e}$ be an n-dimension real vector; $\vec{e}$ can be thought of as a function on the vertices of G. We say that $\vec{e}$ is an *eigenvector* of A with *eigenvalue* λ if and only if $\vec{e}A = \lambda\vec{e}$. It is a well known fact of linear algebra that every $n \times n$ real symmetric matrix A has a *spectrum* of n orthonormal eigenvectors $\vec{e_1}, \vec{e_2}, \ldots, \vec{e_n}$ with real eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ [69, 157]. The eigenvectors are unique up to degeneracies related to equal eigenvalues. In general, the spectral filtering method can be applied with any matrix with real spectrum.

We demonstrate the essence of the spectral method with an example. The left panel of Figure 9 gives the adjacency matrix of a symmetric graph. A dot in position (i, j) in this graph corresponds to a link between i and j. There are two highly connected clusters in this graph; the first includes nodes 1 through 200 and the second all the other nodes. The two clusters are connected with a few links. The right panel of Figure 9 plots the weights assigned by the eigenvector which corresponds to the second largest eigenvalue. The nodes belonging to the first cluster were assigned positive weights and the nodes of the second cluster negative weights. Thus, an efficient heuristic to separate the two clusters is to examine the eigenvector.

In broad lines, the spectral filtering method for an $n \times n$ symmetric matrix A proceeds as follows: STEP 1: Compute the k largest eigenvalues of A together with the corresponding eigenvectors.

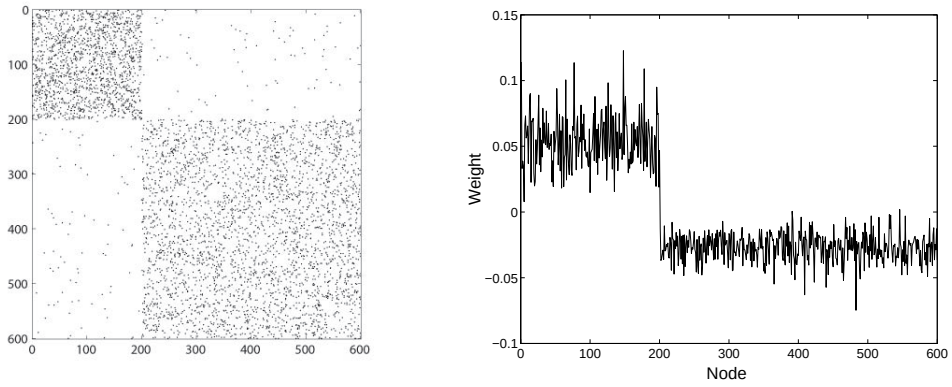


Figure 9: The adjacency matrix (left) of a random graph on 600 vertices. There is a dot in position (i, j) iff there is a link between i and j. The first and second diagonal blocks correspond to subgraphs with high connectivity. Off-diagonal blocks represent sparse edges between the subgraphs. The second eigenvector (right) assigns positive weights to the nodes of the first block and negative weights to the nodes of the second block.

The parameter k depends on the application and the instance, but it is always one to two orders of magnitude smaller than n.

STEP 2: For each i, $1 \leq i \leq k$, let $\vec{e_i}$ be the eigenvector associated with λ_i. Sort the vertices according to the weight assigned by $\vec{e_i}$. A typical profile of the sorted vertices is in Figure 10. Cut towards the most positive end (or towards the most negative end), with special preference to sharp jumps, if they exist. A good example of a sharp jump can be found in Table II. These groups are candidates for clustering and/or semantic proximity. There is no known general rule for determining the cuts; practical heuristics are application specific.

In general, the eigenvectors corresponding to large eigenvalues tend to capture global characteristics of the graph and its semantics, such as groups of nodes $S \subset V$ for which the ratio

$$\frac{\text{edges inside } S}{\text{edges incident to } S} = \frac{|\{(i,j) \in E : i \in S, j \in S\}|}{|\{(i,j) \in E : i \in S, j \in V\}|} \tag{21}$$

is large, indicating *clusters* of relatively high connectivity and, thus, presumably further semantic proximity, not necessarily otherwise expressed in the data (the deep theory of "expander" graphs supporting this claim can be found, for example in the books of Chung [37] and Sinclair [144]). In addition, because there is no polynomial time algorithm to find a set S minimizing the above ratio, the spectral method is an efficient heuristic. Eigenvectors corresponding to small eigenvalues tend to capture noise, or local characteristics that are explicit or can be easily computed from the data.

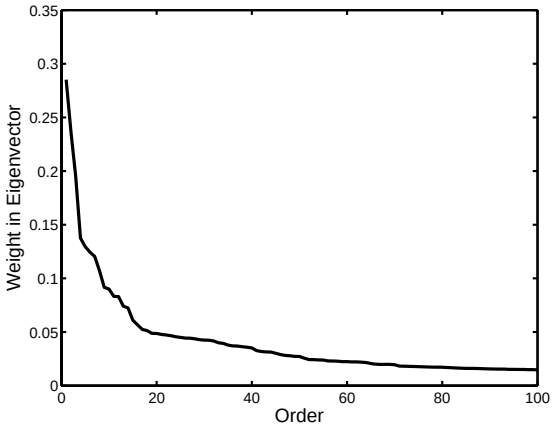


Figure 10: Typical profile of the most positive weights assigned to nodes by the eigenvector corresponding to a large eigenvalue. This profile was taken from a principal eigenvector of the stochastic normalization of the AS topology. Data from Agarwal et al. [4] (10-Feb-2004).

4.2.2 Algebraic Primitives of Spectral Filtering

More formally, we list a few technical facts which build the intuition behind the spectral filtering method (the statements are straightforward, though some of the proofs to which we point are quite technical).

(a) The largest eigenvalue λ_1 of a d-regular graph is d and the corresponding eigenvector assigns uniform weights to all vertices [37, 102]. All other eigenvalues λ_i, $2 \leq i \leq n$ are small, $|\lambda_i| \leq O(\sqrt{d})$, almost surely [37].

(b) The eigenvalues λ_i, $1 \leq i \leq n$, of a graph with m edges and maximum degree d are bounded by $|\lambda_i| \leq \min\{\sqrt{m}, d\}$ [102].

(c) The spectrum of the union of vertex disjoint graphs is the union of their spectra [37, 102].

(d) If A and B are the adjacency matrices of not necessarily disjoint graphs with eigenvalues $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n$ and $\beta_1 \geq \beta_2 \geq \ldots \geq \beta_n$, then the eigenvalues of their union $C = A + B$ are $\gamma_1 \geq \gamma_2 \geq \ldots \geq \gamma_n$ with $\alpha_i + \beta_n \leq \gamma_i \leq \alpha_i + \beta_1$, $1 \leq i \leq n$ [69, 157]. In addition, the corresponding invariant subspaces of C follow from the invariant subspaces of A perturbed by no more than the maximum invariant subspace of B [69, 146].

The intuition behind the spectral filtering method is that, if we take the union of two vertex disjoint regular random graphs A_1 and A_2 and connect them with a few random edges B, then, combining Facts (a) through (d) above, the spectrum of $C = A_1 + A_2 + B$ will have $\gamma_1 \simeq \gamma_2 \simeq d$

(corresponding to the largest eigenvalues of A_1 and A_2) and $\gamma_i \simeq O(\sqrt{d})$, $3 \leq i \leq n$. Furthermore, we expect to identify the vertices of A_1 and A_2 by examining the eigenvectors corresponding to the first two eigenvalues. See Figure 9. Indeed, the second eigenvector assigns mostly large negative weights on A_1 and mostly large positive weights on A_2.

4.2.3 Similarity Transformation $\mathrm{SIM}(A) = A \cdot A^T$

Now suppose that $G(V, E)$, $|V|$=n, is a directed graph, and thus the adjacency matrix A is no longer symmetric. A is no longer guaranteed to have a complete real spectrum, and the notion of clustering is not well defined either. Let A^T be the transpose of A, i.e., $a_{ij}^T = a_{ji}$. Notice that the product $A \cdot A^T$ is a symmetric matrix. Notice further that its (i, j)-th entry is $\sum_{k=1}^{n} a_{ik} a_{jk}$, measuring the number of nodes that i and j point to in common. In the case where the nodes represent ASes and edges are directed from customers to their providers, the above sum relates i and j to the number of their common providers. Similarly, the product $A^T \cdot A$ relates i and j to the number of their common customers. The transformation $A \cdot A^T$ is very common in spectral analysis. Depending on the application, it is called self-adjoint, co-citation, co-variance, or *similarity transformation*. Here we shall use the notation $\mathrm{SIM}(A) = A \cdot A^T$.

4.2.4 Stochastic Normalization

The intuition behind the spectral filtering method that we gave in the previous paragraphs referred to regular graphs. Indeed, in practice, the spectral filtering method has been found to deteriorate rapidly when the frequencies of non-zero entries vary substantially [70], which is certainly the case with the very skewed degrees of Internet topologies. Inverse frequency normalization is a general approach to restore spectral filtering in such cases.

In its simplest form, inverse frequency normalization divides each entry a_{ij} with the sum $\sum_j a_{ij}$ of the entries of the corresponding row, thus obtaining a matrix where all the rows add up to 1. Notice that this is now a stochastic matrix, in the sense that it describes the transition probabilities of a Markov chain in the natural way. If, in addition, we make all diagonal entries $a_{ii} = 1/2$ and multiply all other entries by $1/2$ the range of the eigenvalues shifts to $(0, 1)$[2] . Like symmetric

[2]On the other hand, the eigenvectors of stochastic matrices are not necessarily orthogonal, and sometimes additional

matrices, such stochastic matrices have a complete spectrum of real eigenvalues and eigenvectors. For any matrix A, we denote its stochastic normalization $\mathrm{N}(A)$. In what follows, we may apply the stochastic normalization to either A or $\mathrm{SIM}(A)$, thus getting $\mathrm{N}(A)$ or $\mathrm{N}(\mathrm{SIM}(A))$.

4.2.5 Faloutsos' Eigenvalue Power-Law

Faloutsos et al. examined the spectrum of the adjacency matrix of the AS Internet topology, without performing any normalization or other transformation [52]. They reported a power-law on the twenty or so largest eigenvalues of this matrix with exponent between .45 and .5.

Mihail and Papadimitriou observe that Faloutsos' eigenvalue power-law is a direct consequence of the degree sequence power-law along the lines of Facts (a) through (d) of Section 4.2.2, in the following sense (see also Figure 11) [113]:

STEP 1: Decompose an undirected AS topology A as $A = F + E$, as follows. Initially F is the set of vertices that have the k highest degrees, and let $d_1, d_2, \ldots, d_k$ be these degrees. Initially F contains no edges. Let E be the entire AS topology graph. Now we will remove some edges of E and add them to F, so as to create k disjoint stars in F. We do this by the following process: For each vertex v that is not in F, if v is incident to k_v vertices in F, pick one of these vertices u with probability proportional to the degree of u in the entire graph, make the edge $\{v, u\}$ incident to the vertex $u \in F$ and remove the edge $\{v, u\}$ from E. Notice that F is now a set of vertex disjoint stars with degrees $d'_1, d'_2, \ldots, d'_k$, and E is the initial AS topology where all edges belonging to the stars have been removed.

STEP 2: Notice that the eigenvalues of a star of degree d are $\pm\sqrt{d}$ and 0 with multiplicity d–1 [102]. Thus, by Fact (d) of Section II.A, the largest eigenvalues of F are $\sqrt{d'_1}, \sqrt{d'_2}, \ldots, \sqrt{d'_k}$. Also, by Fact (d) of Section II.A, the largest eigenvalues of $A = F + E$ cannot be perturbed by more than the largest eigenvalues of E.

STEP 3: For typical AS topologies, we have found experimentally that the above procedure, for $k = 100$, gives $d'_i \simeq d_i$, $1 \leq i \leq k$, hence the largest eigenvalues of F are close to $\sqrt{d_1}, \sqrt{d_2}, \ldots, \sqrt{d_k}$, and the largest eigenvalues of E are, in the worst case strictly smaller than $\sqrt{d_1}$ and on the average 1/5 of

normalizations that rectify orthogonality are necessary for good results. In our analysis this did not turn out to be necessary. We also note that there are many further normalization methods, including so-called Laplacians and divisions by logarithmic or other functions of $\sum_j a_{ij}$, but, again, we did not use them in our analysis.

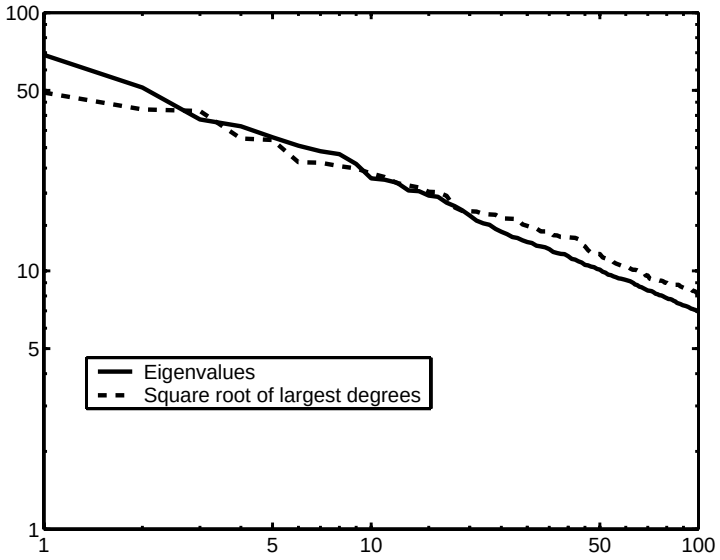


Figure 11: We plot the 100 largest eigenvalues of the adjacency matrix of a typical undirected AS topology and compare them to the square roots of the 100 largest degrees. The eigenvalue power-law follows the degree power-law. Both axes are in log scale. Data from Agarwal et al. [4] (10-Feb-2004).

$\sqrt{d_1}$. Now by Fact (d) of Section II.A, the largest eigenvalues of $A = F + E$ can be understood to be close to $\sqrt{d_1}, \sqrt{d_2}, \ldots, \sqrt{d_k}$. Hence, for graphs where the largest degrees follow Zipf with exponent close to 1, as reported for the AS Internet [52] , the largest eigenvalues follow a power-law with exponent close to .5, also as reported for AS the Internet [52] . We also refer to [35, 53, 56, 68, 113] for formal analysis of these results in stochastic models of power-law random graphs.

We may now conclude that by looking at the eigenvectors corresponding to the largest eigenvalues examined by Faloutsos et al. [52] we should not hope to get information beyond the ASes of highest degree and their customers. Indeed, in experiment, we have found these eigenvectors to be highly concentrated on the large ISPs. Therefore, to obtain more interesting clusters, we will need the processing discussed in Sections 4.2.3 and 4.2.4.

4.3 Spectral Analysis of AS Internet Topology

In this section we describe the spectral analysis that we performed on AS Internet topologies. We discuss the used data, the processing, the behavior of large eigenvalues, and the resulting groups of ASes from the corresponding eigenvectors. We show that clustering varies in the core and the edge of the network, as well as across different geographic areas. On the other hand, the clustering is

consistent over time. Finally, we compare the spectral characteristics of the real AS topologies to synthetic topologies.

4.3.1 Data Used, Transformations and Normalizations

We have used topology data from two sources. The first source is the data of Agarwal et al. [4] who collect BGP routing information from many routers in the Internet and combine all the routing tables to reconstruct the undirected AS topology. Using the heuristics proposed by Subramanian et al. [148], they also provide the information whether an edge of the undirected topology corresponds to a customer-provider or a peering relationship. Finally, Subramanian et al. also give a heuristic to assign the ASes to the levels of a 5-level hierarchy [148]. The most important ASes, such as big ISPs in the core of the Internet, are assigned to level 1. The smallest ASes are assigned to level 5. The topological data from this effort dating on April 6, 2002 are the ones used most in our study[3].

The second set of data is from the National Laboratory for Applied Networkwing Research (NLANR) [121]. Though this data is far less complete, it has the advantage that it spans the time period of 1997 to date. We have thus used this data to study the evolution of clustering over time. The data from NLANR does not contain information about the relationships between the ASes. We have used the algorithm proposed by Gao to infer AS relationships[4] [59].

The data from both sources ([4] and [121]) are not perfectly accurate. We do not believe though that this affects the results of our study, in the sense that missing links would quite likely strengthen the clustering findings.

An AS topology without AS relationships corresponds to an undirected graph with a symmetric adjacency matrix A, in the natural way. For such a topology we perform spectral analysis on the stochastic normalization $\mathrm{N}(A)$. An AS topology with customer-provider or peer relationships corresponds to a directed graph A', where a'_{ij}=1 and a'_{ji}=0 if and only if i is a customer of j and j is a provider of i, and $a'_{ij} = a'_{ji} = 1$ if and only if i and j are peers (in all other cases the entries are 0). For such a topology we perform spectral analysis on the stochastic normalization $\mathrm{N}(\mathrm{SIM}(A'))$.

If we perform spectral analysis starting from the entire undirected graph A or directed graph

[3] We should note that perhaps the most complete set of data is in [33]. It was difficult to annotate this data with the AS hierarchy information of [4], and thus we did not use them.

[4] In addition to customer-provider and peering, Gao includes sibling relationships [59] ; to be consistent with our first set of data, we replace sibling relationships with peering relationships.

A' we find that the clusters indicated by the eigenvectors associated with the large eigenvalues correspond to groups of nodes assigned levels 3, 4 and 5 of the hierarchy proposed by Subramanian et al. [148], thus are away from the core of the network. This is intuitive, since we expect the edge of the network to have more areas with higher connectivity inside the area and relatively lower connectivity to the rest of the network, along ratio Eq. 21 of Section 4.2. Similarly, we expect that the core of the network is better connected, and thus the ratio 21 of Section 4.2 is higher in the core.

To capture the clustering properties of the core of the network we have to explicitly isolate the core from the edge and analyze the core alone. We have used two methods to isolate the core. When information about the AS hierarchy is available, such as in the data of Agarwal et al. [4], we define the core to be the subgraph that contains only the ASes assigned to levels one through four. When the hierarchical information is not available, as in the data from NLANR [121], we iteratively prune all the nodes in the graph that have degree one or two. The graph whose core we wish to find can be either directed or undirected. We denote the core as $\mathrm{Core}(A)$ and $\mathrm{Core}(A')$ depending on whether the original graph was undirected or directed respectively. As above, we perform spectral analysis to $\mathrm{N}(\mathrm{Core}(A))$ and $\mathrm{N}(\mathrm{SIM}(\mathrm{Core}(A')))$.

4.3.2 Results for the Entire AS Topology

Figure 12 shows the largest eigenvalues of the AS topology using data from Agarwal et al. [4]. We have considered the adjacency matrices of the topology with and without AS relationships, for both the entire network and the core. Notice that the eigenvalues are quite high, indicating the existence of clusters in the underlying topology. Another interesting observation is the drop in the eigenvalues between the entire topology and the core of the network. This is expected because the core was constructed by removing small ISP's which tend to cluster more.

Next, we give some representative groups of nodes corresponding to the highest weights assigned by eigenvectors corresponding to large eigenvalues. The first example was taken using the $\mathrm{N}(\mathrm{SIM}(\mathrm{Core}(A')))$. The group corresponds to the largest eigenvalue, which is 1.0. In Table 3, we list the members of the group that take the highest weights in the eigenvector.

In Table 4, we give a group of ASes that belong to Chinese ISP providers. This was taken from the eigenvector of $\mathrm{N}(\mathrm{SIM}(\mathrm{Core}(A')))$ that corresponds to the 6th largest eigenvalue with value

Table 3: A sample of a cluster found in the $N\left(\mathrm{SIM}\left(\mathrm{Core}\left(A'\right)\right)\right)$ topology.

AS	Weight	Description	Country
8075	-0.1298	Microsoft	US
4513	-0.1295	Globix Corporation	US
12956	-0.1078	Telefonica Data Autonomous System	ES
3292	-0.0996	TDC Tele Danmark	DK
3303	-0.0980	Swisscom Ltd	CH
2497	-0.0970	IIJNET	JP
3582	-0.0951	University of Oregon	US
5459	-0.0949	London Internet Exchange Ltd.	GB
6730	-0.0942	sunrise (TDC Switzerland AG)	CH
293	-0.0934	ESnet	US
6079	-0.0909	RCN Backbone	US
3257	-0.0893	Tiscali Intl Network	DE
6461	-0.0888	Abovenet	US
2516	-0.0886	KDDI Corporation	JP
4181	-0.0881	TDS Telecom	US
1668	-0.0868	AOL Transit Data Network	US
3356	-0.0868	Level 3 Communications North America	US
2548	-0.0848	DIGEX-AS	US
8210	-0.0847	Nextra's international backbone	NO
3300	-0.0815	AUCS Communications Services	NL
3549	-0.0805	Globalcrossing	US
2914	-0.0805	Verio	US
1239	-0.0794	SprintLink Backbone	US
701	-0.0787	Alternet	US
6830	-0.0777	CHELLO BROADBAND	NL
5400	-0.0769	BT Ignite European Backbone	NL
3561	-0.0767	Cable & Wireless (CW)	US
1136	-0.0767	KPN OVN IO	NL
2828	-0.0765	XO Communications, Inc.	US
5511	-0.0762	France Telecom	FR

Note: This cluster is taken using the eigenvector which corresponds to the highest eigenvalue. The ASes in this group are big ISP providers, mostly in North America and Europe. The weights of the eigenvector did not show a sharp jump. Data from [4] (10-Feb-2004).

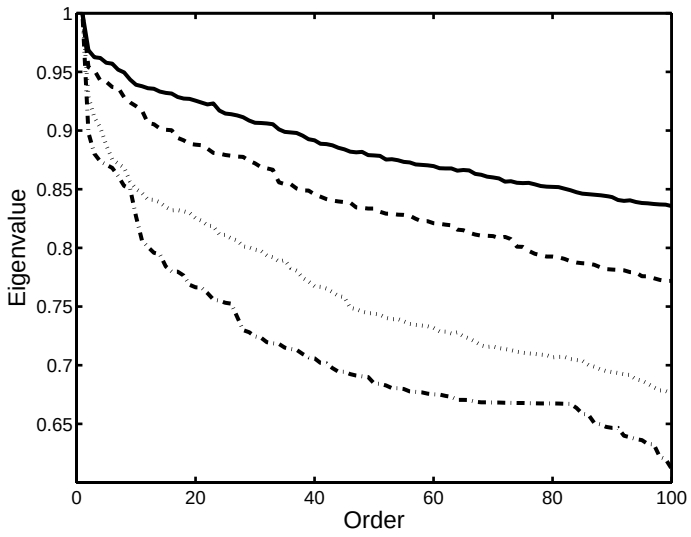


Figure 12: The largest eigenvalues of a typical AS topology. The top line corresponds to the entire topology without AS relationships N(A). The second line corresponds to the entire topology with AS relationships N(SIM(A')). The third line corresponds to the core without AS relationships N(Core(A)). The bottom line corresponds to the core with AS relationships N(SIM(Core(A'))). Data from Agarwal et al. (6-Apr-2002) [4].

0.8363. Notice that the clusters of relatively big ASes in Tables 3 and 4 (levels 1 through 3 of the hierarchy) appear in prominent positions when we examine the core of the topology. As we shall see below, such clusters do not appear when we examine the entire topology.

In Table 5 we give a group of ASes that belong to Greek academic institutions. This was taken from the eigenvector of $\mathrm{N}(\mathrm{SIM}(A'))$ that corresponds to the 2nd eigenvalue with value 0.9539. Notice that this cluster of rather small ASes (levels 4 and 5 of the hierarchy) appears in prominent position when we examine the entire topology.

We should note that the three examples presented here are typical. We chose to include the particular examples wanting to give one cluster from each continent.

4.3.3 Results specific to Geography

Is the Internet topology homogeneous across the entire globe? Do the same connectivity patterns apply everywhere? The first synthetic models of Internet topologies which emphasized the principle of preferential connectivity proposed by Jamin et al. [73] and by Medina et al. [108] were implicitly making such homogeneity assumptions. Recently, these assumptions have been challenged by

Table 4: A sample of a cluster found in the $N\left(\mathrm{SIM}\left(\mathrm{Core}\left(A'\right)\right)\right)$ topology

AS	Weight	Description	Country
9815	0.2684	Beijing Gold First System Engineering Co.	CN
17964	0.2652	Beijing Dian-Xin-Tong Network Technologies Co.	CN
9305	0.2563	Beijing Feihua Communication Technology Co	CN
9802	0.2527	21vianet(China) Inc.	CN
9812	0.2172	Shanghai Cable Network Co., Ltd	CN
9811	0.2145	Srit corp.	CN
4808	0.1749	Chinanet Beijing Site	CN
17431	0.1740	Beijing TONEK Information Technology Development	CN
23610	0.1686	HangZhou City NetCom	CN
9308	0.1630	21VIANET(CHINA)	CN
4837	0.1514	chinanet IDC center beijing node	CN
9929	0.1513	China Netcom	CN
9809	0.1378	New Era Foundation System Co.	CN
17779	0.1375	Shanghai Symphony Telecommunications Co.	CN
17621	0.1372	China Netcom Corp.	CN
7497	0.1332	China Science and Technology Network	CN
9800	0.1235	CHINA UNICOM	CN

Note: This group was found in the eigenvector corresponding to the 16th largest eigenvalue. Data from [4] (10-Feb-2004).

Table 5: A sample of a cluster found in the $N\left(\mathrm{SIM}\left(\mathrm{A}'\right)\right)$

AS	Weight	Description	Country
6802	-0.1949	National Educational and Research Information Network	BG
8522	-0.1840	Foundation for Research and Technology	GR
13092	-0.1840	Univerzitet u Beogradu	YU
5379	-0.1799	FYROM Academic and Research NETwork	MK
15536	-0.1790	European Centre for the Development of Vocational Training	GR
15948	-0.1697	ICE/HT	GR
8214	-0.1697	Albania National Education Research Network	AL
8248	-0.1697	Greek High-School Internet Network	GR
12364	-0.1697	University of FYROM	MK
12402	-0.1697	University of Piraeus	GR
3323	-0.1697	National Technical University of Athens	GR
5470	-0.1697	Aristotle University of Thessaloniki	GR
6744	-0.1697	Academic & Research Network in the Region of Patras	GR
6867	-0.1697	University of Crete	GR
8618	-0.1697	Ionion University	GR
9069	-0.1697	Technological Educational Institute of Athens	GR
8253	-0.1697	Democritus University of Thrace Network	GR
8278	-0.1697	Technical University of Crete	GR
8643	-0.1697	Academic and Research Network in the Region Athens	GR
8700	-0.1697	TECHNOLOGICAL EDUCATIONAL INSTITUTE OF LARISSA	GR
8762	-0.1697	Technological Educational Institute of Crete	GR
8991	-0.1697	Institute of Marine Biology of CRETE	GR
20551	-0.1697	Technological Educational Institute (T.E.I.) of Patras	GR
20813	-0.1697	Hellenic Open University	GR
8581	-0.1697	University of Ioannina	GR
2546	-0.1697	Greek Academic & Research Computer Network	GR
5489	-0.1697	T.E.I. of Thessaloniki	GR
8611	-0.1697	Athens University of Economics and Business	GR
8617	-0.1697	University of the Aegean	GR
15690	-0.1697	National Observatory of Athens	GR
5408	-0.1683	Greek Research and Technology Network	GR
8880	-0.0349	AII/Capital IT Network , 9 Ivan Vazov Str.	BG

Note: The whole group contains several more ASes related mostly to academic institutions in Greece and occasional ASes from other Balkan countries. This group was found in the eigenvector corresponding to the 2nd largest eigenvalue. Data from [4] (10-Feb-2004).

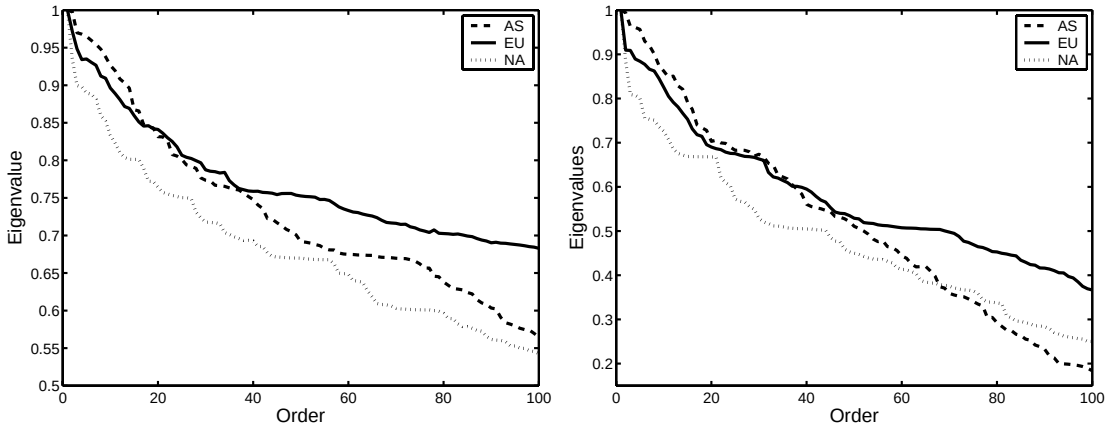


Figure 13: The spectrum of different continents. The top graph is for the entire topology of each continent, while the bottom graph is for the core of the topology of each continent. Data from Agarwal et al. (6-Apr-2002) [4].

Lakhina et al. [91] and Yook et al. [159] who show strong correlation between the placement of ASes and routers with geography as well as economic development. We second and strengthen these findings, by observing that different geographic parts of the network exhibit different connectivity patterns.

We have used data from the NetGeo project [28] to assign ASes to continents. We constructed three graphs for the continents of North America (NA), Europe (EU) and Asia (AS)[5]. We included AS relationships, thus obtaining non-symmetric adjacency matrices $A'_{\rm NA}$ for North America, $A'_{\rm EU}$ for Europe and $A'_{\rm AS}$ for Asia. In Figure 13 we give the largest eigenvalues of $\mathrm{N}(\mathrm{SIM}(A'_{\rm NA}))$, $\mathrm{N}(\mathrm{SIM}(A'_{\rm EU}))$ and $\mathrm{N}(\mathrm{SIM}(A'_{\rm AS}))$. We also give the plots for the spectrum of the corresponding cores $\mathrm{N}(\mathrm{SIM}(\mathrm{Core}(A'_{\rm NA})))$, $\mathrm{N}(\mathrm{SIM}(\mathrm{Core}(A'_{\rm EU})))$ and $\mathrm{N}(\mathrm{SIM}(\mathrm{Core}(A'_{\rm AS})))$. The point to notice is that, both in the entire topology and in the core, North America exhibits less clustering than Europe and Asia. This can be understood intuitively by thinking of the network in North America as being at a later evolutionary stage, and hence is more connected.

4.3.4 Spectrum Consistency over Time

Is the spectral behavior of the Internet topology consistent over time? See Figure 14. We have used data from NLANR [121] taken one year apart and found consistent behavior of the largest

[5]It is possible that some ASes are present in more than one continents. We treated such ASes as belonging to only one continent. However, their number is very small, and the results are not affected.

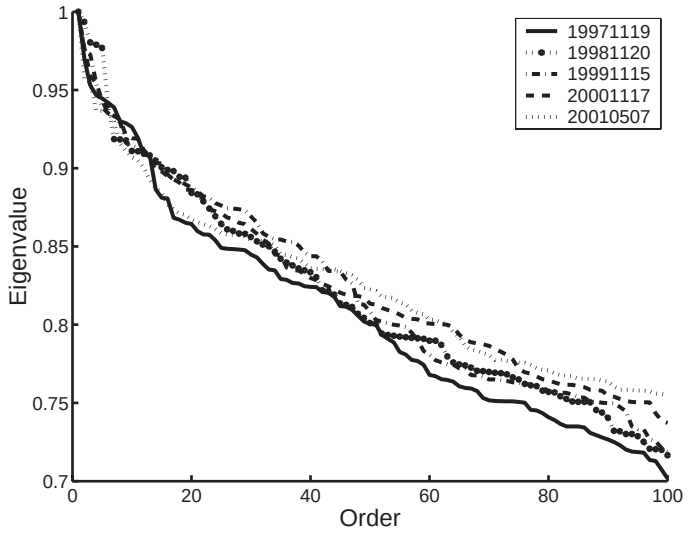


Figure 14: The evolution of the largest eigenvalues of the AS topology. This data is from NLANR [121].

eigenvalues of N(A). This confirms the intuitive belief that the spectrum is a robust characteristic of a topology. Figure 14 refers to the entire AS topology without AS relationships. We have observed similar behavior in the evolution of the AS topology with AS relationships, as well as the core of the topology, and when restricted to specific continents.

4.3.5 Synthetic topologies

In Figure 15 we give the largest eigenvalues of the AS Internet topology, as well as similar graphs generated by Inet [73], Waxman, growth with preferential connectivity according to Barabasi-Albert and the improved GLP heuristics [27, 108] which explicitly tries to capture better clustering (all the above for the same number of nodes as the Internet topology), and the power law random graph (PLRG) model of Aiello et al. [5] (for the specific degree sequence of the Internet topology). We give the spectrum of both the entire AS topology and the core (recall that the core of synthetic topologies where there is no other indication of hierarchy is obtained by iterative pruning).

For the entire Internet topology, all synthetic generators, except for the Power Law Random Graph (PLRG) [5], have smaller eigenvalues. This means that they do not contain as strong clusters as the real Internet. This could have been expected since no synthetic generator attempts to capture such explicit notions as geography and business interests. But, why is PLRG an exception? Note

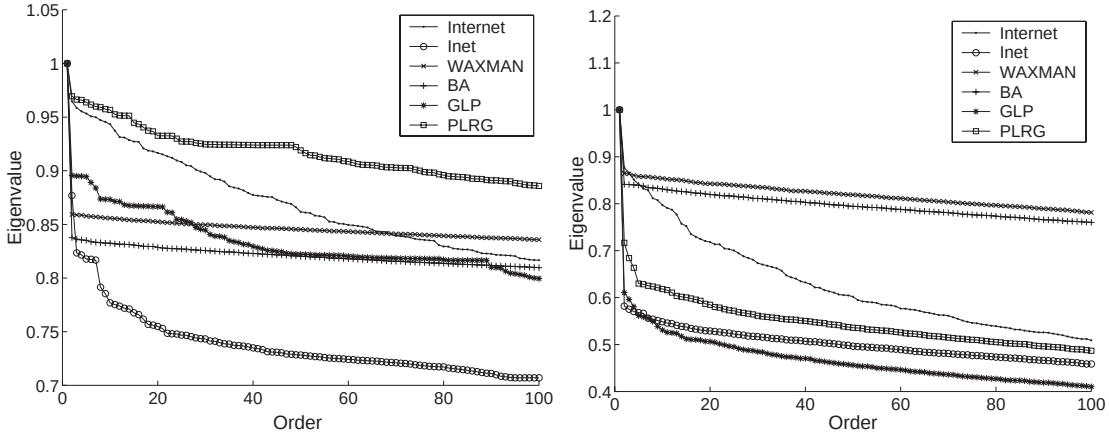


Figure 15: Spectrum of real and synthetic Internet topologies. The top graph corresponds to the entire topology. The bottom graph corresponds to the core. Data for the Internet AS topology from Agarwal et al. (13-Nov-2003) [4] . All synthetic topologies have approximately the same number of nodes as the Internet AS topology.

that PLRG does not even generate a connected graph [5]. So, the same random principles that generate several isolated connected components in the entire graph, generate several badly connected subgraphs within the giant connected component.

For the core of the topologies, the WAXMAN and BA models produce higher eigenvalues. We believe that this is a pathological byproduct that these topology generators do not attempt to simulate any notion of core. Therefore, the behavior of the spectrum after pruning small degree vertices is the same as the entire topology.

For further comparative purposes, we give the evolution of the largest eigenvalues of the stochastic normalizations of synthetic topologies (Figure 16).

4.4 Impact of Spectral Analysis on Performance and Traffic Primitives

What is the significance of the information retrieved by the spectral analysis of Section 4.3? What is the significance of the eigenvectors associated with the large eigenvalues? The main difficulty in answering this question is in deciding which metric to pick and examine its correlation with clustering. In general, there is no consensus on the metrics by which Internet topologies should be evaluated. One approach is to include detailed graph properties [27, 73, 108], while another approach is to use metrics that distinguish graphs with heavy tailed degree sequences as opposed to more regular topologies and may be correlated with further coarse characteristics of the network [135, 150]. Our

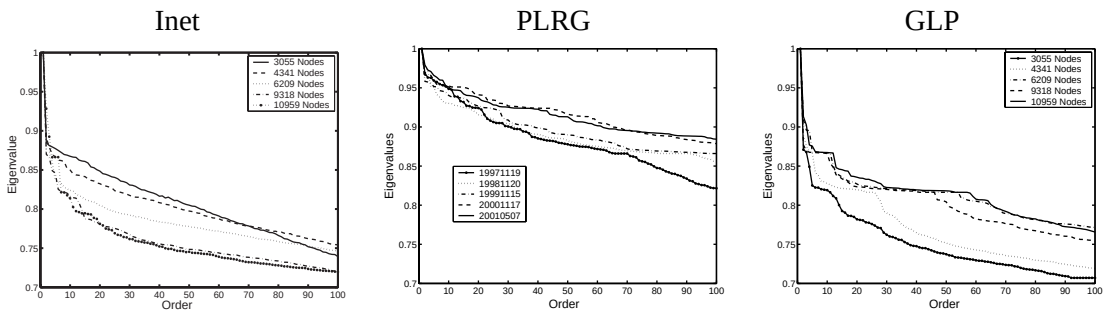


Figure 16: Evolution of the largest eigenvalues of the stochastic normalization of topologies generated with the Inet, PLRG, and GLP topology generators. The topologies generated with Inet and GLP have the same number of nodes as the topologies in Figure 14. PLRG used as input the degree sequences of the topologies of Figure 14. It appears that PLRG has the smallest drift over time, thus resembling the real Internet of Figure 14. On the other hand, the spectrum of Inet shifts down, while the spectrum of GLP shifts up.

approach is closer to the latter, and influenced from the proposals of Fabrikant et al. [51] and Carlson et al. [31] that *topology properties should be studied in connection to the functionality of the network*. In particular, we shall study the correlation of the information retrieved from the eigenvectors of Section 4.3 to the performance of a primitive experiment that studies the "congestion" in the network.

For an undirected (without AS relationships) topology, suppose that we send one unit of traffic along a minimum hop (shortest) path from each node to every other node[6]. This induces a *stress for each link* defined as the total number of paths going though the link. We study the maximum link stress, which can be thought of as an indicator of congestion.

In an intra-cluster traffic scenario, we expect that there is more traffic between ASes that have geographic or business relationships. We use the following spectral-filtering based heuristic to group ASes into clusters:

(a) If n is the size of the topology, consider the $\alpha \cdot n$ largest eigenvalues of $N(A)$, and the eigenvectors associated with each such eigenvalue. In our experiments, we have used $\alpha = .5$.

(b) Consider the nodes H_1 and H_2 that are assigned the highest $\beta \cdot n$ positive and the highest $\beta \cdot n$ negative weights in each such eigenvector. The parameter β is set to $.25$ in our experiments.

(c) Each AS which appears in H_1 or H_2 for at least one examined eigenvector will be assigned to

[6]In case of many shortest paths, we pick one of them arbitrarily.

the cluster of the positive or negative end of the first eigenvector in whose H_1 or H_2 it appeared. In this way we assign ASes to at most one cluster.

We say that a *traffic pattern is* $\epsilon\%$ *intra-clustered* if each node sends $\epsilon\%$ of its traffic exclusively inside the cluster that it belongs, and $1-\epsilon\%$ of its traffic uniformly to all nodes (thus uniform traffic is 0% intra-clustered).

We are interested in studying the change in the max link load as the traffic shifts from *uniform* to *intra-cluster*, and *inter-cluster*. It is reasonable to expect that, in general, topologies with higher principal eigenvalues, and thus worse cuts (in the sense of (1) of Section 4.2), should tend to exhibit worse link stress behavior. Thus, as we shift traffic from uniform to intra-cluster (resp. inter-cluster), we expect the maximum link stress to drop (resp. increase) significantly, since we are increasing (resp. decreasing) the traffic that stays inside the cluster and reducing (resp. increasing) the traffic that crosses the bad cut.

Indeed, the AS Internet topology is exhibiting sharper shift in link stress behavior than several synthetic topologies from Brite (BA,GLP,Waxman [15, 27, 108, 156]), Inet [73], and PLRG [5]. The results are given in Table 6. Assume for example that the traffic is 20% intra-clustered. Then, the maximum link stress for the AS topology dropped to 91.5% of that in uniform traffic. For the same intra-cluster traffic, the max link stress in the topology generated by Inet dropped to 97.7%. Thus, the maximum link stress decreased by a factor of 8.5% in the case of the AS topology and by 2.3% in the case of Inet. At the extreme of 100% intra-clustered traffic the max link stress in the Internet drops by more than 40%, while in every synthetic topology the drop was less than 23%, with the exception of Waxman, in which case the drop was around 30%.

Observe from Table 6 that the transit-stub model of GT-ITM [30] behaves similarly to the Internet AS topology with respect to the changes to the maximum link stress as the traffic pattern changes. For the purposes of this study, we forget that the GT-ITM produces a router-level topology. Instead we focus our study on the properties of the generated graph. The two-layer transit-stub model, by design, generates topologies with clusters, where each cluster is composed of the routers that belong to the same network. We view the fact that our clustered-driven traffic patterns cause sharper results in GT-ITM, which explicitly produces clustered graphs, as further confirmation of our proposed method.

Table 6: Drop of max link stress as the traffic shifts from uniform to intra-cluster and inter-cluster.

A. Intra-cluster

	Internet	Inet	PLRG	GLP	BA	Waxman	Transit-Stub
0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
20%	91.5%	97.7%	95.6%	95.8%	96.4%	94.1%	92.2%
40%	83.0%	95.4%	91.2%	91.6%	92.9%	88.2%	84.4%
60%	74.4%	93.1%	86.9%	87.3%	89.3%	82.3%	76.6%
80%	65.9%	90.8%	82.5%	83.1%	85.8%	76.3%	68.8%
100%	57.4%	88.5%	78.1%	78.9%	82.2%	70.4%	61.0%

B. Inter-cluster

	Internet	Inet	PLRG	GLP	BA	Waxman	Transit-Stub
0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
20%	108.5%	102.4%	102.5%	103.9%	102.8%	107.6%	107.8%
40%	116.9%	104.7%	105.1%	107.8%	104.7%	116.1%	115.6%
60%	125.4%	107.1%	107.6%	111.7%	107.1%	124.6%	123.4%
80%	133.8%	109.5%	110.1%	115.6%	109.5%	131.2%	131.2%
100%	142.3%	111.8%	112.7%	119.5%	111.8%	141.7%	139.0%

Note: The AS Internet exhibits more drop than any synthetic topology (almost twice as much with the exception of the Waxman and Transit-Stub models). We note that these numbers refer to the core of the network. The behavior was similar when we did the same experiment in the whole network, and in each specific continent.

We therefore propose that the information retrieved from the eigenvectors associated with the largest eigenvalues may be suggestive of intra-cluster traffic patterns. We propose to use the clusters suggested by these eigenvectors as one meaningful way to generate traffic patterns that deviate from uniform traffic. One additional remark is due. It may be thought that the decrease in link stress under intra-cluster traffic patterns is a straightforward consequence of shorter min-hop paths that would be used in an intra-clustered traffic pattern. See Table 7. For each node, define its *expected hop distance* as the expected hop distance of the node from every other node under a specific traffic pattern. Notice that both in the Internet and in the synthetic topology produced by Inet, the drop in the average expected hop distance is not nearly as striking as that of the max link stress. We therefore conclude that the drop in the link stress is a result of a *different distribution* of traffic over shortest paths, rather than a mere decrease in their lengths. Thus the intra-cluster traffic pattern is indeed non trivial. Similar observation apply to inter-cluster traffic patterns.

Table 7: Drop in max link stress and average expected hop distance, as the traffic shifts from uniform to intra-clustered.

	Internet Max Link Stress	Internet Avg. Exp. Hop Dist	Inet Max Link Stress	Inet Avg. Exp. Hop Dist
0%	100.0%	3.3744	100.0%	2.7499
20%	91.5%	3.2855	97.7%	2.7151
40%	83.0%	3.1965	95.4%	2.6802
60%	74.4%	3.1076	93.1%	2.6454
80%	65.9%	3.0187	90.8%	2.6106
100%	57.4%	2.9297	88.5%	2.5757

Note: (a) The same trend applies to the other synthetic topologies. (b) Observe that the average path length between any two nodes does not change significantly.

4.5 Ranking by the First Eigenvector

The "significance" of an AS, or its position in a hierarchy, is a subjective matter, in the sense that ASes are never explicitly or implicitly assigned such rankings. There is relatively good agreement about the "top" and "bottom" of a hierarchy. For example, an ISP that has only peers and no provider is almost surely very big, while an AS that has no customers or peers and only one or two providers is almost surely very small. In two separate efforts, Gao [59] and Subramanian et al. [148] gave heuristics to assign hierarchical levels to ASes, after inferring AS relationships and taking into account several non-trivial further characteristics.

In this Section we observe that a different heuristic, based on the weights assigned to the ASes by the first eigenvector of a suitably defined modification of the directed AS graph (i.e., after AS relationships have been inferred), is highly correlated with the results of the hierarchy algorithm proposed by Subramanian et al. [148].

The proposed heuristic is an adaptation of the *pagerank* method used by Google to infer quality of Web pages [124]. The analogy is natural. Both the directed AS topology and the WWW are directed graphs. In the WWW, a hyperlink pointing from page i to page j indicates an endorsement of importance from i to j. In the Internet, an edge pointing from customer i to provider j can be thought of as a similar endorsement of importance, while in peers the endorsement becomes mutual.

The ranking method is the following. Let A' be the directed adjacency matrix. For each node i define the outdegree of i as $d_{\text{out}}(i) = |\{j : a_{ij} = 1\}|$. Now consider the stochastic matrix

$$\mathrm{P}(A) \ : \ p_{ij} = \begin{cases} \frac{\alpha}{d_{\text{out}}(i)} + \frac{1-\alpha}{n} & \text{if } a_{ij} = 1 \\ \frac{1-\alpha}{n} & \text{if } a_{ij} = 0 \end{cases}$$

The above stochastic matrix represents a random walk on the directed graph A', where with probability α we go to a provider or peer chosen uniformly at random, and with probability $1-\alpha$ we jump to a uniformly random node from the set of all nodes (the latter step is a standard correction to avoid degeneracies pertaining to sinks).

Let $\pi(v)$ be the stationary probability of the stochastic matrix $P(A)$. Google assigns to Web pages pagerank quality $\pi(v)$. By analogy, we assign to each AS hierarchical weight $\pi(v)$. In Figure 17 we compare the results of the hierarchy algorithm proposed by Subramanian et al. [148] to our hierarchical weight $\pi(v)$. We have used $\alpha = .95$; the results are similar for any $.9 \leq \alpha \leq .99$. To plot the graph, we have grouped the ASes by their level in the hierarchy. Then, we sort the ASes in each group by their weight in $\pi(v)$ and plot the weights in decreasing order. Observe that we use logarithmic scale for both axes.

There is notable correlation between the weights assigned to the ASes and their level in the hierarchy. Nodes assigned in high levels in the algorithm of Subramanian et al. [148] have higher values in $\pi(v)$. Also, the weights assigned to the ASes of a group are in general higher than the weights assigned to ASes that belong in groups of lower level. One noticeable exception is the weights assigned to levels 4 and 5. ASes in these levels have very small degrees and they cannot be separated by the page rank method. At first glance it seems that there is an "anomaly" in the figure, since there are some ASes that are assigned larger weights than ASes which belong to higher levels. We argue that this could be a problem of the subjective nature of hierarchical assignment, and/or the heuristic used by Subramanian et al. [148] to assign ASes to levels. We will discuss two examples to make this point. The largest weights in levels 2 and 3 have a very high value which is comparable to the weights assigned to nodes in level 1. These weights correspond to the ASes of Tiscali Intl Network (AS number 3257) and of Abovenet (AS number 6461) respectively. We believe that they had to be assigned in the highest level. This is justified by their degrees in the adjacency matrix, which are 330 and 585 respectively, and by the reputation they have as big ISP providers.

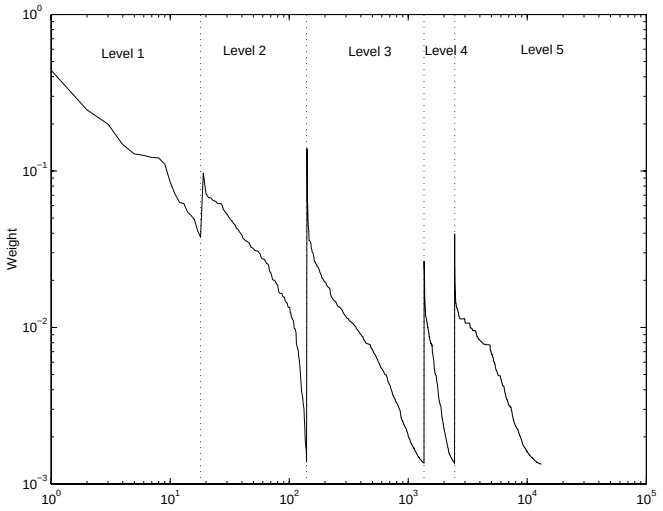


Figure 17: Comparison of hierarchy with the first eigenvector.

We extend the above method to obtain an assignment of significance to links. If n is the number of ASes and m is the number of links of the undirected AS topology, let $N{=}n^2$ be the number of pairs of ASes and associate with each such pair a shortest path between their endpoints. We may now consider the $m{\times}N$ traffic routing matrix T, where each column corresponds to a shortest path and there is a 1 on the rows of the links used by the path. Such matrices are common in Network Tomography [39, 153]. If there are multiple shortest paths, we choose one arbitrarily. Using the SVD method, which is a generalization of the decomposition into eigenvalues and eigenvectors for non-square matrices, we can compute the left eigenvector of T that corresponds to the largest eigenvalue. Just like pagerank, this eigenvector gives an order of importance to links. Links that get higher values are associated with links that accept more traffic and thus are candidates to be places of congestion. Observe that this statement was made without making any assumption about the traffic between any two ASes.

To find the correlation between the importance assigned to links and the amount of traffic they receive we did the following experiment[7]. We assumed that between each pair of ASes there is some amount of traffic flowing drawn from a uniform distribution that takes values between 0 and 2

[7]For this example we have used an induced graph of the real topology which includes all the ASes in levels 1 and 2 as assigned by the algorithm of Subramanian et al. [148]. Memory and processing limitations did not allow us to work with bigger matrices.

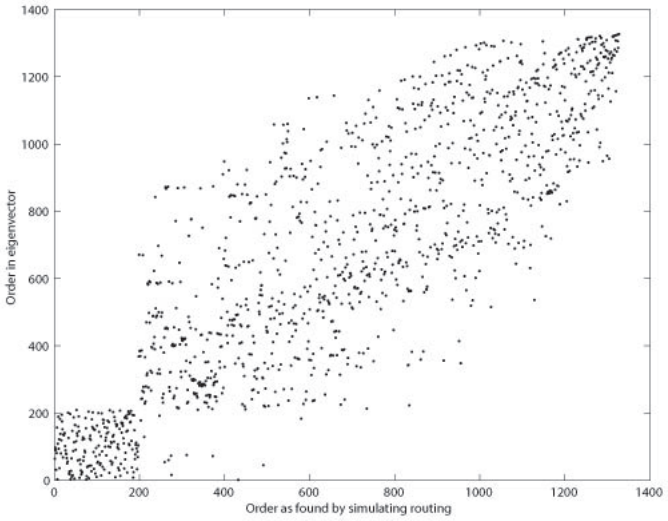


Figure 18: Correlation between link importance as assigned by the left eigenvector of the SVD of the traffic matrix with the load of the link. Correlation coefficient is 0.8594.

traffic units[8]. After performing shortest path routing and assigning loads to links, we have ordered the links by their load. We are interested to find the relation between this ordering and the ordering given by the weights in the eigenvector. In Figure 18 we depict this relationship. There is a point in (i, j) when a link is in i-th position sorted by the load and in j position sorted by the weight in the eigenvector. Indeed it is easy to observe that there is strong correlation between the importance of the link and the amount of traffic it receives. The correlation coefficient in this case is 0.8594 indicating this strong correlation.

In addition, it is possible to use the left eigenvectors to identify *clusters* of related links that form a cut in the original adjacency matrix. The links in the cut carry traffic between areas in the Internet that are not well connected and thus they are candidates to be points of congestion. As a simple example we give Figure 19, where we draw a cluster of links (cluster in the same sense as the clusters defined earlier for ASes) taken from the left eigenvector which corresponds to the second largest eigenvalue. Intuitively, we expect that indeed the trans-atlantic links to carry a lot of traffic and thus be points of congestion as indicated. We have observed similar clusters using the other eigenvectors, and also in positions that seem intuitively natural (across Central and Eastern

[8]Setting the traffic to 1 for each pair gave the same results.

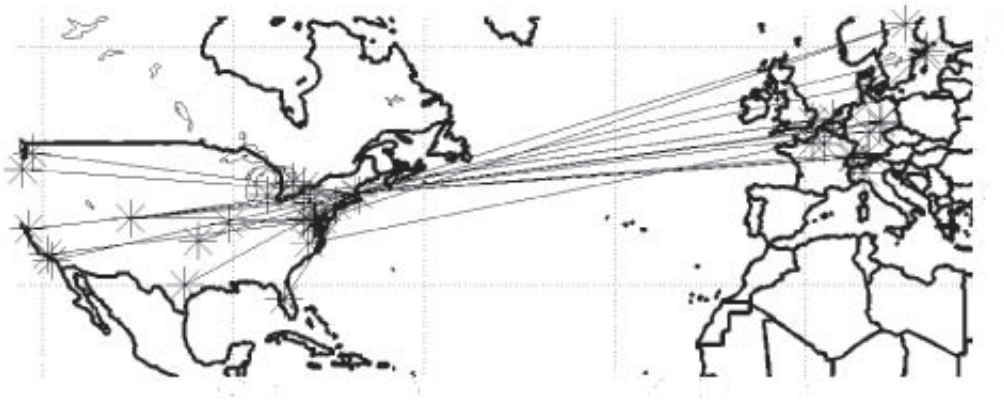


Figure 19: An example of a link cluster.

Europe, across the Pacific, e.t.c). It is still an open question to us how the clusters observed in the AS topology relate with the link clusters.

CHAPTER V

SEARCHING AND TOPOLOGY CONSTRUCTION IN PEER-TO-PEER NETWORKS

5.1 Introduction

The simulation of a random walk, or more generally a Markov chain, is a fundamental algorithmic paradigm with profound impact in algorithms and complexity theory. Furthermore, it has found a wide range of applications in such diverse fields as statistics, physics, artificial intelligence, vision, population dynamics, bioinformatics, among others.

Recently, random walks have been proposed as primary algorithmic ingredients in protocols addressing searching, topology maintenance, and in gossip protocols of unstructured P2P networks. In particular: (a) Following extensive experimentation, Lv et al. report that searching by simulating random walks is preferable to the standard practice of searching by flooding [105]. They attribute the suitability of random walks to their adaptivity in termination conditions and hence granularity in coverage of the search space (in flooding, increasing the time-to-live (TTL) by 1 may increase the space coverage exponentially). (b) Law and Siu give a distributed algorithm for constructing and maintaining unstructured topologies with very strong connectivity properties, namely constant degree and constant expansion, with $O(\log n)$ overhead per addition of a peer, where n is the number of peers [92]. At a very high level, when a new peer arrives, one would ideally attach the new node to existing peers chosen uniformly at random. The protocol of [92] approximates such uniform sampling by simulating $O(\log n)$ steps of a random walk. (c) Bawa et al. [16, 17], Kempe et al. [79] and McSherry and Kempe [80] introduce aggregate functions (min, max, sum, count, avg) as fundamental computational tools in P2P protocols, and appeal to the simulation of random walks as natural schemes to implement these computations.

What are the analytic reasons of the success of the random walk method? Can we isolate one or two comprehensible analytic primitives that explain the power of the method? Most important,

can we translate these primitives to heuristics, or rules of thumb, for the use of the method in P2P network applications?

Independent sampling from the uniform distribution is a primodal statistical and algorithmic primitive. However, it is infeasible to implement in many populations of complex systems, such as the set of nodes of a P2P network. The difficulty arises from the fact that this set is not centrally maintained and it is also quite dynamic. In this chapter we make the following arguments: **(i)** The random walk method is an excellent candidate to simulate sampling for P2P networks. **(ii)** The number of simulation steps required can be as low as the number of samples in independent uniform sampling, which translates to constant network overhead, independent of the size of the network. In particular, beyond termination adaptivity and space coverage granularity discussed in [105], we believe that the power of the random walk method can be pinned down into two kinds of analytic properties: The first analytic property, corresponding to **(i)**, is as follows. Consider a population whose members can be connected by links forming a connected graph. Perform a random walk starting at any state and simulate the random walk for τ steps. Use the state of the random walk at state τ as a sample point. This simulates sampling with arbitrary accuracy, for τ bounded by well defined parameters of the graph. The above is rather intuitive, since we expect that graphs without any "bad" cuts will make the walk to "lose memory" and hence reach a random state quite fast. In particular, for a wide range of applications, it is possible to construct sparse graphs so that $\tau = O(\log n)$. Such fast convergence rates translate to practically efficient simulation of uniform sampling. In the theoretical computer science this property is known as rapid mixing [118, 144]. In the context of P2P networks [16,17,79,80,92,105] also appeal (directly or indirectly) to rapid mixing for the efficiency of their protocols. The second analytic property, related to **(ii)**, is substantially more profound and rather counter-intuitive. It states that starting the random walk at a random state, simulating the walk for k steps, and using each visited node as a sample point, we may achieve same statistical properties as k independent uniform samples. The reason why this is counter-intuitive is because of the obvious huge dependencies between successive steps of a random walk.

In this chapter we focus on two central issues of P2P networks. The first issue is searching and computing averages; these are closely related. The second issue is overlay topology construction. For both problems we have isolated scenarios where independent uniform sampling would

have been a good algorithmic primitive. We use k successive states visited during the simulation of a random walk on the P2P network in the place of k independent samples. We compare the performance of sampling by random walk to the performance of the previously known methods for search and construction. For search our performance metric is the number of items found for a given number of query messages; for analytical clarity we do not consider further metrics such as response time or reliability (of course, the random walk method which is inherently sequential incurs larger response times than flooding and, in practice, we expect hybrid schemes such as parallel random walkers or random walks with local floodings [2, 64, 105, 115, 128, 142]). Our results and conclusions are:

For searching relatively popular items as well as for computing averages, we found, experimentally, that random walk performs better than flooding, for the same number of network messages, in two cases. The first case is when peers in the topology form clusters so that the whole topology is arranged in two tiers, the lower representing peer clustering and the higher connecting representatives from each cluster (e.g. super-nodes) to ensure good global connectivity. The second case is when the same search request is re-issued repeatedly, in hopes of finding new peers, while the entire topology has not changed dramatically (less than 40%). We believe that both scenarios are realistic. However, to the best of our knowledge, they have not been considered in previous studies. Thus, we believe that our performance evaluation framework is an additional original contribution of our work. Our results in the context of searching and computing averages are intuitive. Our primary contribution was to formalize set-ups of practical interest and translate our analytic intuition in these set-ups. From the technical standpoint all these results are derived from the good expansion properties of the underlying random graph models, including the case of clustering.

For constructing and maintaining a P2P topology with good connectivity properties, we turned to the approach of [92]. As mentioned before, when a new peer arrives, [92] find a few nearly uniformly random existing peers to connect the new peer by simulating $O(\log n)$ steps of a random walk. This causes $O(\log n)$ network overhead per newly arriving peer. We introduce a daemon construction and connect a newly arriving peer with constant network overhead. Our construction is based on the second analytic property, described earlier in this section. In fact, there are strong dependencies between the edges of peers that arrive closely in time. We give analytic evidence

that these local dependencies do not affect the global connectivity properties of the network, up to constant factors. We also give strong experimental evidence that our daemon construction simulates the algorithm of [92] (and other related constructions) with overhead a very small constant per arriving peer, for networks up to 5M nodes (which is the current believed size of Kazaa; larger experiments were stressing the memory limits of our machines), and with truly negligible penalty in the quality of the connectivity of the overall topology. We believe that our results in the context of P2P network construction are particularly surprising. From a practical point of view, they indicate that minimal amount of correctly used randomization suffices to keep a dynamic network well connected; however, the actual implementational details of our scheme are beyond the scope of our work. From a theoretical point of view, we believe that our results lead to an exciting new paradigm in the study of the power and necessity of randomness. All of our algorithms in the context of construction were inspired by the second analytic property of random walks.

The isolation of the second property has been one of the most celebrated results in complexity theory [7, 10, 61, 72]. In complexity theory this property has been used as follows. Consider a randomized algorithm that uses n random bits and has probability of success 1/2. By simulating the algorithm k times we may decrease the failure probability to $1/2^k$. This needs kn random bits. Now think of the nodes of a graph labeled by all 2^n n-bit strings that can be used to simulate a randomized algorithm. Consider a constant degree "expander" graph $\mathcal{H}$ imposed on these 2^n nodes (there are known deterministic constructions of such expander graphs [58]). Start from a uniformly random point of $\mathcal{H}$ and simulate a random walk on $\mathcal{H}$ for k steps. This requires $O(n)$ random bits to pick the initial point, and $O(k)$ bits to perform the walk. Then simulate the randomized algorithm on the k n-bit strings visited by the random walk. The failure probability is $O(1/2^k)$, and yet we used only $O(n+k)$ random bits to perform the experiment! This idea is the basis of several pseudorandom number generators with provably good performance. In some sense, our work can be viewed as translating a complexity result, mostly known in the context of savings in random bits, into savings in overhead and improved performance in a practical networking context.

The balance of the chapter is as follows. In Section 5.2, we give the supporting theory, comparing coupon collection and Chernoff bounds to the corresponding statements for random walks. All the technical parts of this section are known; their synthesis and relevance in the context of

networking is new. In Section 5.3, we use random walks to perform searching in P2P networks and compare this approach to searching using flooding and uniform sampling. We also discuss random walks as primitives to compute average aggregates. In Section 5.4, we describe two algorithms for distributed construction of P2P topologies with good expansion properties. We stress that our experiments in Sections 5.3 and 5.4 are on the size of current P2P networks. Many implementational and other details are suppressed to emphasize clearly the new ideas.

5.2 Statistical Estimation and Random Walks

In this section we focus on the statistical properties of sampling performed by ideal independent draws, and compare them to the statistical properties of sampling performed by simulating a random walk. In the case of independent sampling we are interested in the number of samples sufficient to achieve a certain statistical property. In the case of random walks we are interested in the number of simulation steps sufficient to achieve the same statistical property. For comparison, we consider two common abstractions, namely the coupon collection problem and Chernoff bounds for independent Bernoulli trials; these abstractions refer to sampling by independent draws. For sampling by random walk simulation, we consider the cover time which is the suitable analogy to coupon collection, and the trajectory sample average which is the suitable analogy to independent Bernoulli trials. We observe that, for both abstractions, the overhead of the random walk simulation method is determined only by the second eigenvalue of the probability transition matrix of the random walk. In particular, when the second eigenvalue is constant (independent of the size of the graph) the random walk method achieves the same statistical characteristics as independent sampling, up to $O()$ notation. The reason why the second eigenvalue provides such clean characterizations is that it is intimately related to global connectivity properties of the graph, namely expansion and conductance. Intuitively, expansion and conductance express the worst-case cuts of the graph, and it is natural to expect that when the graph does not have bad cuts a random walks approaches its stationary distribution very fast, and hence sampling by random walk mimics independent sampling well.

5.2.1 Coupon Collection and Chernoff Bounds

The coupon collection problem is the following: Suppose that there are n distinct types of coupons. At each step, we draw a coupon whose type is uniformly distributed among all n types. Let T_n be the time by which we have encountered coupons belonging to all n distinct types. It is well known [54] that

$$E[T_n] = 1 + \frac{n}{n-1} + \frac{n}{n-2} + \ldots + n = O(n \log n) \ . \tag{22}$$

Let δ be a constant, $0 < \delta < 1$. Let $T_{\delta n}$ be the time by which we have encountered coupons belonging to δn distinct types. It is also well known that

$$E[T_{\delta n}] = 1 + \frac{n}{n-1} + \ldots + \frac{n}{n-\delta n+1} = \frac{1}{1-\delta} O(n) \ . \tag{23}$$

We proceed with an outline of Chernoff bounds [13]. Let $X_1, \ldots, X_k$ be independent Bernoulli trials with $\Pr[X_i = 1] = p$ and $\Pr[X_i = 0] = 1-p$, $0 \leq p \leq 1$, $1 \leq i \leq k$. Let $X = \sum_{i=1}^{k} X_i$, hence $E[X] = kp$. In a searching context, where p denotes the probability that a randomly drawn object has a desired property, we are interested in the probability that the property is found in substantially fewer draws than its frequency in the search space. This corresponds to the event $X \leq (1-\epsilon)kp$, for $0 < \epsilon < 1$. For this event, Chernoff bounds are $\Pr[X \leq (1-\epsilon)kp] \leq e^{-\frac{\epsilon kp}{2}}$. In a measurement context, where p denotes the fraction of objects satisfying a certain property, we are interested in the quality of the estimator X/k for p. Now, for $0 < \epsilon < 0.9$, Chernoff bounds are:

$$\Pr\left[|\frac{X}{k} - p| \geq \epsilon p \right] \leq 2e^{-\frac{\epsilon^2 kp}{20}} \ . \tag{24}$$

5.2.2 Random Walks, Convergence, Cover Time and Trajectory Sample Average

Let $G(V, E)$ be an undirected connected graph, $|V| = n$. Let d_i denote the degree of vertex i, $1 \leq i \leq n$. Let $d_{\min} = \min_{1 \leq i \leq n}\{d_i\}$. Let $A = \{a_{ij}\}$, $1 \leq i, j \leq n$, be the adjacency matrix of G. Let P be the transition matrix of the random walk on G, where a particle that is on vertex i at time t, moves to a neighbor of i at time $t+1$, chosen uniformly at random among all neighbors of i. It is well known and easy to verify that the above random walk has a unique stationary distribution $\vec{\pi}$, in the sense that $\vec{\pi}P = \vec{\pi}$, with $\pi_i = d_i/2|E|$, $1 \leq i \leq n$, and let $\pi_{\min} = d_{\min}/2|E|$. Now let f be a 0-1 function on V, $f : V \xrightarrow{f} \{0, 1\}$. Let p be the probability mass of vertices that take the

value 1 under f under the stationary distribution $\vec{\pi}$, which is the same as the mean of f under $\vec{\pi}$: $p = \sum_{v \in V: f(v)=1} \pi_v = \sum_{v \in V} f(v)\pi_v$.

Of particular interest are the following three metrics: (a) Convergence rate, which is the rate with which the random walk approaches the stationary distribution. (b) Cover time, which is the time when the random walk has visited all vertices at least once. This is analogous to the coupon collection abstraction, and we wish to have bounds comparable to (22) and (23). (c) Trajectory sample average, which is the rate with which the value of f, averaged over successive vertices of a trajectory of the random walk, approaches p. This is analogous to the Chernoff bound abstraction, and we wish to have bounds comparable to (24). In the next paragraph we point out bounds for all the above metrics in terms of the second eigenvalue of P.

5.2.3 Bounds in terms of the Second Eigenvalue

In general, a vector $\vec{x}$ is an eigenvector of P with eigenvalue λ if and only if $\vec{x}P = \lambda\vec{x}$. Thus, $\vec{\pi}$ is an eigenvector of P with eigenvalue 1. It is well known that P has n real eigenvectors with corresponding eigenvalues $1 = \lambda_1 > \lambda_2 \geq \ldots \geq \lambda_n \geq -1$ [69, 157]; the strict separation of the first and second eigenvalues follows from the connectivity of G. The first eigenvalue $\lambda_1 = 1$ which corresponds to eigenvector $\vec{\pi}$ characterizes stationarity. We may also assume that $|\lambda_2| > |\lambda_n|$ (large negative eigenvalues concern strong periodicities, like bipartiteness, which we may exclude for the purposes of this work). Consider a random walk on G according to the transition matrix P, starting from an arbitrary vertex, or an arbitrary distribution on V. Let y_t be the vertex that the random walk visits at time t, $1 \leq t \leq \infty$. To bound the convergence rate of the random walk we focus on the so-called variation distance which, at time t, is $\Delta(t) = \max_{S \subset V} |\Pr[y_t \in S] - \vec{\pi}(S)|$. The following is known [144]:

$$\Delta(t) \leq \pi_{\min}^{-1} \lambda_2^t \ . \tag{25}$$

Let C_n be the time by which the above random walk visits all the vertices of G. [11, 24] show:

$$\begin{aligned} E[C_n] &\leq O\left(\pi_{\min}^{-1} \log n/(1-\lambda_2)\right) \\ &= O\left(n \log n/(1-\lambda_2)\right) \ , \ \text{for } \pi_{\min} = \Omega(\tfrac{1}{n}). \end{aligned} \tag{26}$$

Compare (26) to (22) and realize that, for constant λ_2, they both solve coupon collection in the same order of magnitude. Let $C_{\delta n}$ be the time by which the random walk visits δn distinct vertices of G,

for some constant δ, $0 < \alpha < 1$. It is straightforward to derive from [11, 24] (and is folklore among probabilists) that

$$\begin{aligned} E[C_{\delta n}] &\leq \tfrac{1}{1-\delta} O\left(\pi_{\min}^{-1}/(1-\lambda_2)\right) \\ &= \tfrac{1}{1-\delta} O\left(n/(1-\lambda_2)\right) \ , \ \text{for } \pi_{\min} = \Omega(\tfrac{1}{n}). \end{aligned} \tag{27}$$

Compare (27) to (23) and realize that, for constant λ_2, they both solve partial coupon collection in the same order of magnitude.

Recall that y_t denotes the vertex that the random walk visits at time t. Let $Y_t = f(y_t)$. Suppose that we simulate the random walk for

$$\tau = \frac{\log \pi_{\min}^{-1}}{1-\lambda_2} \geq \frac{\log \pi_{\min}^{-1}}{\log \lambda_2^{-1}} \tag{28}$$

steps. Very roughly, this guarantees good approach to stationarity according to the bound on $\Delta(t)$. Then use the next k steps as sample points. We thus let $Y = \sum_{t=\tau+1}^{\tau+k} Y_t$. The particularly strong result is that, using Y/k as an estimator for p, is of the same quality as Chernoff bounds. Thus, despite the local dependencies introduced by consecutive steps of the random walk, the overall distribution of the vertices visited by the random walk is well spread across the sample space. In particular, (29) below is to be compared to (24):

$$\Pr[|\frac{Y}{k} - p| \geq \epsilon p] \leq 8e^{-\frac{\epsilon^2 k p^2 (1-\lambda_2)}{20}} \ . \tag{29}$$

The above result was obtained (in increasingly stronger forms and referring to pseudorandom number simulation) in a sequence of celebrated complexity theory papers [7, 10, 61, 72]. The version that we give above is from [61].

5.2.4 Second Eigenvalue, Expansion and Conductance

For $S \subseteq V$, define the cutset of S, $C(S)$, as the set of edges with one endpoint in S and the other endpoint is $\bar{S}$. Define the volume of S as the sum of the degrees of vertices in S: $\mathrm{vol}(S) = \sum_{v \in S} d_v$. The edge expansion, γ, and the conductance, Φ, of G are measures of the connectivity of G, where the cut sizes are normalized by the size of the partition classes in which they break the graph (intuitively higher γ and Φ indicate that for any partition of nodes into two clusters there are many edges that connect the two clusters). Formally, edge expansion γ and conductance Φ are

defined as follows:

$$\gamma = \min_{\substack{S \subset V \\ |S| \leq |V|/2}} \frac{|C(S)|}{|S|} \ , \ \Phi = \min_{\substack{S \subset V \\ \mathrm{vol}(S) \leq \mathrm{vol}(V)/2}} \frac{|C(S)|}{\mathrm{vol}(S)}. \tag{30}$$

In addition, the following bound is known [144]:

$$1 - 2\Phi \leq \lambda_2 \leq 1 - \frac{\Phi^2}{2} \ . \tag{31}$$

Finally, realize that in graphs where we have bounds on the minimum and maximum degrees, both expansion, conductance and eigenvalues are easily related. In particular, in a family of regular graphs, if any of these metrics is a constant then all of them are constants. In summary, for families of graphs where λ_2 is constant, consecutive states of random walks are excellent candidates to approximate independent uniform sampling. Since, λ_2 constant is equivalent to expansion ϕ constant, and constant expansion is equivalent to good global connectivity, it is reasonable to try the random walk approach in communication networks. Good global connectivity is desired and believed to hold in all reasonable networks and network models [25, 36, 48, 62, 114, 125].

5.3 Searching and Computing Aggregates

In this section we study the performance of searching using flooding and random walks, and compare the two methods to each other and to a baseline case of independent uniform sampling. We measure the performance in terms of the average number of distinct copies of an item located in the search, the probability of not finding any copy of the item, and the number of messages that the searching algorithm uses. We show experimentally that searching by random walk is better than flooding, if at least one of the following conditions holds: The first condition is that there is peer clustering. That is, there are communities in the topology, with dense connectivity between peers in the same community and sparse connectivity between peers of different communities. The second condition is that the user issues multiple search requests for the same item and between two consecutive requests the topology changes relatively slowly (in the sense that two consecutive snapshots of the topology are highly correlated). We believe that the two scenarios introduced above are important for the reasons listed below. For both these scenarios the effectiveness of the random

walk follows by the good connectivity (measured by λ_2 and conductance) of the underlying random graph models. In further work [23, 64] discuss how the random walk method adapts when the underlying topology has bad cuts, e.g., by biasing the walk to cross these clusters.

In practice, when a user issues a request, the user (or, the system on behalf of the user) re-issues the same request multiple times hoping to locate more sources. Consecutive floodings take advantage of the changes in the topology so as to discover more sources. If the topology however remains mostly unchanged between consecutive requests, then the new floodings will mostly discover sources already known. On the other hand, for the same number of messages, the random walk follows totally different trajectories and has better chances to discover new sources. Note that in previous work, searching has been always modeled as an one time process. However, we believe that studying searching under multiple requests and a changing topology is realistic and important.

The motivation behind studying peer clustering becomes clear if we consider the process by which the P2P network is formed. Each peer keeps a cache of other peers and picks its neighbors from its cache. The cache is populated by the addresses of peers that answered previous queries [122]. Thus, intuitively, the cache contains addresses of peers that have similar interests. It is therefore reasonable to expect that this process leads to the formation of communities of users. The exact process by which P2P networks are formed is largely unknown and thus peer clustering is at this point only a hypothesis. But, we believe that it is a fair hypothesis based both on our practical experience with P2P systems, and on the observation that most networks grown in a decentralized way exhibit strong clustering properties. See also [27, 60, 66, 90] and for related discussion [93].

The rest of the section is organized as follows. In paragraph 5.3.1 we give the methodology. In paragraph 5.3.2 we discuss the most simple case of a topology without clustering that does not change with time. In this scenario, we find that flooding and random walk behave similarly. In paragraph 5.3.3 we examine topologies with peer clustering. In paragraph 5.3.4 we examine multiple re-issues of a request in topologies that change with time. In paragraph 5.3.5 we discuss power-law graphs and real topologies. Random walks behave better than flooding in most cases of interest. Finally, in paragraph 5.3.6 we discuss the efficiency of computing averages by a deterministic distributed algorithm reminiscent of the spread of a probability distribution by a random walk.

5.3.1 Methodology

In all our experiments we assume that copies of the item to be discovered populate $\alpha\%$ of the peers, where α is a parameter with $0.01\% \leq \alpha \leq 0.1\%$; this represents items that are not rare. Assuming that a single search reaches 10000 distinct nodes, a typical value of the horizon of a user in the Gnutella network [122], the search will result, in expectation, in 1 to 10 distinct copies found for the range of α's we have experimented with. In our experiments we have used parameters that will result in querying around 10-30K nodes. When comparing different algorithms we measure their relative performance for the same number of queries (10-30K queries).

The performance of each searching technique is measured as the number of distinct copies located when simulating the searching algorithm from a randomly chosen peer of the topology. To make statistically robust conclusions, we repeat the experiment from a set of randomly chosen peers, typically 500 peers, and study the distribution of the number of distinct copies located (hits).

A remark is due for the chosen range of α and the chosen range for the number of queries. The chosen range of α which represents relatively popular items allows us to measure, in addition to simple success or failure, detailed statistics, such as mean and standard deviation described below. We found similar results for searching very rare items, for example a single copy in the entire network, but we do not discuss these results since the only result we can report is success or failure.

The above results hold for the chosen range of queries (10-30K). We chose this range because, as discussed above, it is realistic. Searching larger portions of the network gives similar results up to a constant fraction of the nodes of the network (for example, 50%). If we increase the number of queries so that the entire network can be discovered then flooding performs better. By (22) and (26) random walks will use $O(n \log n)$ queries, while flooding will use $O(m)$, where m is the number of edges.

Performance Metrics: A metric that summarizes the distribution of the hits is the mean. Of equal importance, however, is the discrepancy around the mean, and the failure probability (probability of no copies discovered). Even when the means of random walks and flooding are the same, it is almost always the case that the discrepancy and failure probability of random walks are substantially better than flooding (e.g. see Figure 21). We therefore measure "Mean", standard deviation "Std", and

"Failure probability".

Cost: We measure the cost of each searching technique as the number of messages or queries performed during the search. When comparing different algorithms, it is always under the assumption of using the same number of messages. In the case of random walks, the number of messages is simply the number of steps of the random walk. In the case of flooding, the number of messages sent during the searching is as follows: Recall that upon arrival of a query from a neighbor, a node forwards the query to all other neighbors while decreasing the TTL by one, and thus generates as many messages as its neighbors minus one, unless the same query has been received previously, or the TTL on the query has reached zero.

Peer-to-peer topologies: Available topologies of current P2P networks are limited in size and of questionable quality due to the collection method (topologies from [100] have only 30-40K nodes, when current P2P networks have hundreds of thousands and perhaps millions of users). We therefore experiment on synthetic topologies of up to 1 million nodes. Experimenting with extremal synthetic topologies has the additional advantage of facilitating the demonstration of general principles.

We have used the following models to generate synthetic P2P topologies:

- Flat regular expanders. This is a canonical example of regular graphs with good expansion properties. We use expanders since expansion is desired and believed to hold in every reasonable network and network model [36, 62, 92]. We have used 6-regular expanders.

- Two-tier topologies with clustering. To study the effects of peer clustering we have started by constructing a number of isolated regular expanders that correspond to the clusters. Then, from each cluster we pick a small number of nodes at random and connect them using another regular expander.

- Power-law graphs. Many important networks that arise in a decentralized fashion are known to have power-laws [48, 52, 90]. Some researches argue that P2P topologies may also possess heavy tails [74]. We have used the standard model of growth with preferential connectivity to generate power-law random graphs (this model runs in linear time and hence can efficiently generate graphs of very large size).

- Samples of real topologies. We have used partial views of the Gnutella topology made available in [100]. These topologies are limited in the number of peers (around 35K) and of questionable accuracy, since the topology evolves during the topology discovery process, some peers are uncooperative and for other practical reasons. Because of their very limited size, our results are inconclusive.

Dynamic Topologies: The dynamic nature of P2P topologies is a crucial parameter of these systems [32]. However, very little is known about the way these topologies evolve over time. To model the dynamic nature of the P2P topologies we have used the following heuristic: We perform a number of "rewirings"; for each rewiring we pick two edges uniformly at random and exchange their end points. The number of rewirings is a parameter that is related to the speed by which the topology is changing. In our experiments, the measurements are happening before and after the rewirings and not during the process of changing the topology. The size of the topology remains unchanged during the rewirings, because we want to capture only the effects of changes in the connectivity and not in the number of peers.

In our experiments we measure the speed by which the topology is changing as the ratio of the number of links changed by performing rewirings over the total number of links. We have experimented with ratios in the range from 2% to 40%. The rate of change in current P2P networks is not known and difficult to estimate. However, from our practical experience, we observe that consecutive searches happening every 10-20min do not result in large differences in hosts discovered, as would have been expected if a large fraction of the network has changed.

Content placement: The straightforward approach is to pick the nodes that will host the copies uniformly at random from the entire population. This is what we have used in our experiments. We have also experimented with cases where the nodes that host the item are close to each other in the topology (content clustering). In our experiments, we have observed that content clustering affects the performance of searching by flooding or random walk much less than peer clustering, or re-issuing of the same request. We therefore do not present this case.

Table 8: Performance of searching in a static topology without peer clustering.

Attribute	Flooding	RW	Uniform
Mean	8.712	8.796	10.990
Std	3.01	2.93	3.22
Min	1	2	3
Messages	22331	22331	22331
Unique peers	17235	17431	21839

Note: The topology used had 500K peers, $\alpha = 0.05\%$. Min is the minimum number of hits over all searching requests. Unique peers is the number of distinct peers discovered during the search. Observe that flooding and random walk have very similar performance, while uniform sampling is better.

5.3.2 Flat Topologies with Uniformly Distributed Content

We start by giving a scenario in which the performance of flooding and random walks is similar. See Table 8. We study the performance of issuing a request only once in a flat regular topology of 500K peers. After simulating the flooding algorithm with a TTL of 5 and counting the number of messages, we run the random walk algorithm and configured it to use the same number of messages. Observe that the mean and minimum numbers of hits, as well as the standard deviation of the hits distribution of both flooding and random walk are similar, while independent uniform sampling is better. Moreover the entire distribution of hits, given in Figure 20, is similar for both random walk and flooding. We have experimented with topologies of various sizes and for various popularities of files, with $0.01\% \leq \alpha \leq 0.1\%$, and found that always the performance of flooding and random walk are similar, when both are allowed to use the same number of messages.

Observe that compared to the study of [105] we use only one walker, that is, one long random walk. The results were similar when using a larger number of walkers assuming that the total number of messages (sum of the lengths or all random walks) stays the same. The use of more walkers decreases the user-perceived delay, which is a parameter that we do not study in this chapter (see Chapter 6).

5.3.3 Topologies with Peer Clustering

We now examine a topology with well separated communities of peers and show that the random walk method has better performance than flooding. The example topology is constructed as follows.

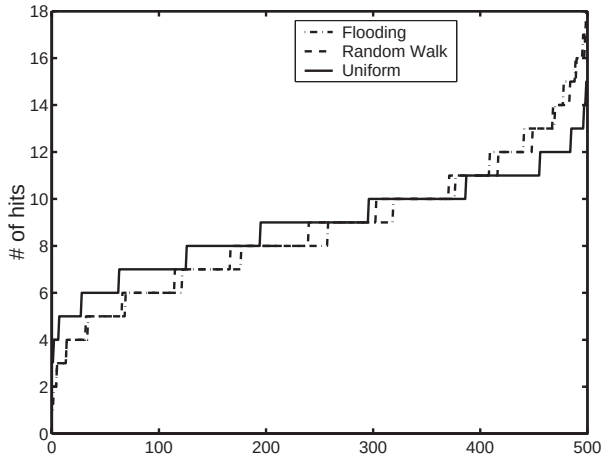


Figure 20: Sorted number of hits when searching from 500 randomly chosen peers. The topology used had 500K peers, $\alpha = 0.05\%$. Observe that flooding and random walk have very similar performance.

We generate five flat regular graphs, each of size 40K. From each topology we pick 1000 nodes at random (for a total of 5K nodes) and construct another flat regular graph on the selected nodes. The final P2P network is the union of all topologies.

The performance of each searching algorithm for different content popularities is given in Table 9. First, observe that the average number of hits using flooding is slightly worse than using random walks but, given the large standard deviations, it could be argued that the differences are not statistically significant. Even though the two schemes behave similarly with respect to the average number of hits, they have totally different behavior when it comes to the failure rate and the minimum number of copies found. For $\alpha = 0.01\%$, flooding failed in 28.8% of the cases and random walk in only 10.8%, an improvement of nearly a factor of 2.

The fundamental difference between the two searching algorithms is that the number of hits in the case of random walks appears more concentrated around the mean (observe also the entire distribution of hits in Figure 21). Indeed, the standard deviation in the case of random walks is much smaller compared to the standard deviation of flooding. Obviously, the optimal concentration around the mean is achieved by uniform sampling. The basic strength of random walks, following Section 5.2, is precisely that they resemble uniform sampling in a quantifiable way.

Table 9: Performance of searching in a topology with peer clustering.

Method	$\alpha = 0.01\%$				$\alpha = 0.05\%$				$\alpha = 0.10\%$			
	Mean	Std	Min	Failure	Mean	Std	Min	Failure	Mean	Std	Min	Failure
Flooding	1.754	1.91	0	28.8%	9.308	6.44	1	0.0%	18.192	11.04	5	0.0%
RW	2.124	1.38	0	10.8%	10.860	3.21	2	0.0%	21.764	4.54	10	0.0%
Uniform	2.676	1.52	0	6.6%	13.496	3.26	5	0.0%	27.274	4.87	15	0.0%

Note: Topology of five clusters for a total of 200K peers. $\lambda_2 = 0.956$.

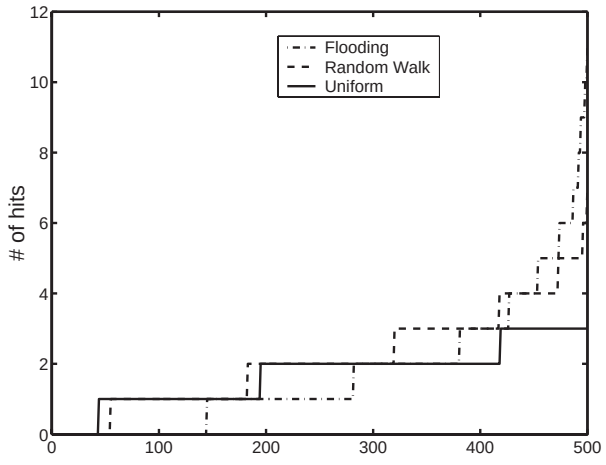


Figure 21: Sorted number of hits in a topology with peer clustering. The distribution of random walk is more concentrated around the mean. Topology of 200K peers, $\alpha = 0.05\%$.

5.3.4 Re-issuing the Same Query

We proceed to study the performance of flooding and random walks in flat regular topologies under the assumption that users issue the same query multiple times while the topology is changing. Realize that dynamic topologies favor only flooding, while random walks are largely unaffected. In fact, in the worst case where the topology is not changing, re-issuing a query k times, by flooding, does not find new copies. On the other hand, according to (27) and (29), increasing the length of the random walk by a factor of k has substantial impact.

The performance of the different algorithms for topologies of various sizes and for various values of α is given in Table 10. The reported experiments are as follows: Each peer initiates a searching request and waits for the results. Then, we change the topology by performing rewiring operations to 2% of the links. Then, each peer initiates a new searching request. We repeat the process four times, simulating consecutive queries. In the end, we count the number of distinct items found for each peer. Table 10 indicates that random walks have better performance compared to flooding with respect to both the average number of hits and the probability of failure. The average number of hits for random walks was at least three times better compared to the same number for flooding. Also, the failure probability dropped substantially. This great performance improvement was expected since, even though we change a certain number of links, the overall topology remains relatively stable and successive flooding searches do not result in many new items found. On the other hand, prolonging the random walk, or, successive random walks from the same peer, follow totally different sampling paths and have better chances of locating new copies of the requested item.

The performance of successive searches depends on the number of topology changes that take place between the consecutive searches. We have studied this effect and report the results in Table 11. We observe that the performance of flooding increases as the rate of topological changes increases. For very fast rates of change (40% at each step in our experiment), the performance of flooding becomes comparable to that of random walks, since effectively the neighborhood of each node changes almost completely between consecutive searches. On the other hand, the performance of random walk remains relatively unaffected by the changes in the topology.

Table 10: Performance of searching in dynamic topologies.

A. Flooding

Size (K)	$\alpha = 0.01\%$				$\alpha = 0.05\%$				$\alpha = 0.10\%$				λ_2
	Mean	Std	Min	Failure	Mean	Std	Min	Failure	Mean	Std	Min	Failure	
100	0.488	0.67	0	60.6%	2.294	1.45	0	8.6%	4.566	2.08	0	1.4%	0.79
300	0.412	0.64	0	66.2%	2.350	1.57	0	9.4%	4.918	2.26	0	0.4%	0.79
500	0.550	0.75	0	58.6%	2.562	1.68	0	8.8%	4.992	2.27	0	0.6%	0.79
1000	0.494	0.62	0	60.0%	2.684	1.72	0	8.7%	5.032	2.31	0	0.2%	0.80

B. Random Walk

Size (K)	$\alpha = 0.01\%$				$\alpha = 0.05\%$				$\alpha = 0.10\%$				λ_2
	Mean	Std	Min	Failure	Mean	Std	Min	Failure	Mean	Std	Min	Failure	
100	1.398	1.14	0	24.6%	7.058	2.40	1	0.0%	14.076	3.30	5	0.0%	0.79
300	1.436	1.12	0	20.2%	7.396	2.62	1	0.0%	14.894	3.87	5	0.0%	0.79
500	1.562	1.19	0	19.8%	7.634	2.78	1	0.0%	15.152	3.88	7	0.0%	0.79
1000	1.518	1.20	0	22.4%	7.544	2.71	1	0.0%	14.982	3.91	4	0.0%	0.80

C. Uniform Sampling

Size (K)	$\alpha = 0.01\%$				$\alpha = 0.05\%$				$\alpha = 0.10\%$				λ_2
	Mean	Std	Min	Failure	Mean	Std	Min	Failure	Mean	Std	Min	Failure	
100	1.734	1.18	0	13.8%	8.634	2.83	1	0.0%	17.210	3.75	7	0.0%	0.79
300	1.864	1.23	0	12.6%	9.212	2.90	1	0.0%	18.496	4.07	9	0.0%	0.79
500	1.872	1.38	0	15.2%	9.486	2.98	2	0.0%	18.924	4.21	10	0.0%	0.79
1000	1.876	1.34	0	14.2%	9.482	3.14	0	0.2%	18.850	4.34	8	0.0%	0.80

Table 11: Performance of searching in dynamic topologies as a function of the rate of changes.

Links	Flooding			Random Walk		
Changed	Mean	Std	Failure	Mean	Std	Failure
2%	0.488	0.67	60.6%	1.398	1.14	24.6%
4%	0.644	0.82	53.0%	1.382	1.11	23.6%
10%	0.888	0.86	38.0%	1.450	1.11	21.4%
20%	1.162	0.99	27.6%	1.456	1.12	20.8%
40%	1.460	1.12	20.0%	1.378	1.13	23.8%

5.3.5 Real topologies and topologies with power-law statistics

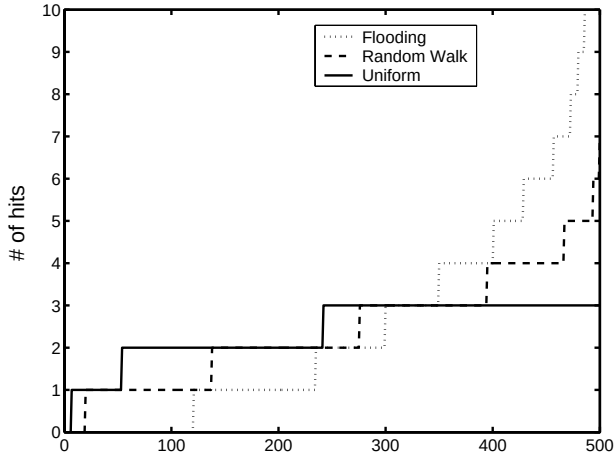
In the previous paragraphs we have experimented on regular graphs. Similar results hold for topologies with heavy-tailed statistics as well as real topologies. In Figure 22, we show the distribution of hits for a topology of 500K nodes generated with the model of growth with preferential connectivity [15], and for a real Gnutella topology taken from [100]. Again, observe that the distribution of hits in the case of random walk is more concentrated around the mean compared to flooding. Indeed, in the real topology, the mean number of hits, the standard deviation and the failure rate were 0.514, 2.15 and 81% respectively in the case of flooding in the real topology, and 0.538, 0.73 and 59% in the case of random walk. Similar results apply for the graph grown with preferential connectivity. Observe that the very small TTL used for flooding in the case of the real topology was due to the small size of the topology. Increasing the TTL to 3, would have resulted in reaching almost half of the nodes with flooding and would have skewed the statistics. It is more realistic to expect that searching visits only a small portion of the graph.

In conclusion, we expect that our results apply to all graphs with good expansion property as expected from the theoretical section. In addition, we expect that typical networks, like P2P networks, have good expansion properties; otherwise, they would not have scaled easily from a few tens of thousands to a few million nodes.

5.3.6 Aggregate Computation

In this paragraph we turn our attention to the efficiency by which the nodes of a P2P network can collectively compute basic aggregates. It has been argued elsewhere that the computation of such aggregates constitute fundamental algorithmic primitives in P2P networks [16, 17, 79, 80]. Here we examine the canonical example of averaging. Let $G(V, E)$ be a graph denoting the network, and let $|V| = n$. Let us assume for simplicity that G is a d-regular graph (the general case is treated in [79]). Recall that, in a typical unstructured P2P scenario, the number of nodes n is not known. Let A be a large number, known to all nodes of the network. Let $\vec{\mu}_t$, $t = 0, 1, \ldots$, be a function on the nodes of the network, with $\mu_0(v_0) = A$, and $\mu_0(v) = 0$ for all $v \neq v_0$. That is, at time $t = 0$, the node v_0 has value A, while all the other nodes have value 0. The goal is to distribute A over the entire network, so that each node has the average value $\eta = A/n$. If this distribution can be achieved efficiently then, for

A. Growth with preferential connectivity model with 500K nodes. (TTL=4)



B. Sample from the Gnutella network with 36K nodes. (TTL=2)

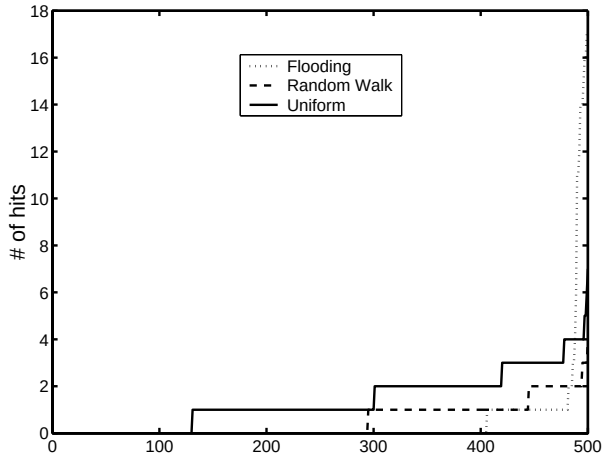


Figure 22: Performance of searching in (A) a network with heavy-tailed statistics, and (B) in a real topology. The small TTL in the real topology is due to the fact that using a larger flooding TTL would have resulted in reaching almost half of the nodes, which is unrealistic.

example, a node can use the value A/η as an estimate for n. Gossip protocols propose that interested nodes perform independent sampling of the network for the purposes of computing η and, in turn, suggest random walks to simulate independent sampling [17, 79]. An even more sophisticated approach proposed in [107] is to have the entire network, collectively, simulate the computation performed by a random walk as follows: Recall that P is the transition matrix of a random walk on G (see Section 5.2.2). The vector $\vec{\mu}_0 P^t$ converges to a uniform vector where each node has value $\mu_t(v) = \eta = A/n$, as $t \to \infty$, (see Section 5.2.2). Indeed, by the bound (25) on the variation distance, we get that $\max_{v \in V} |\mu_v(t) - \eta| \leq n\lambda_2^t$, which means that the difference of the estimate $\mu_v(t)$ from η can be bounded by n^{-c} after $t \simeq (c+1)\log n / \log \lambda_2^{-1}$ simulation steps. Now the interesting observation is that the network can carry out the computation $\vec{\mu}_0 P^t$ step-by-step as follows. Given $\vec{\mu}_t$, the new vector $\vec{\mu}_{t+1}$ is computed when each node v breaks its value $\mu_t(v)$ into d equal parts $\mu_t(v)/d$ and forwards each part to each one of its d neighbors. In turn, v will receive a value $\mu_t(u)/d$ from each one of its neighbors u with $(u, v) \in E$, and will compute $\mu_{t+1}(v) = \sum_{u:(u,v)\in E} \mu_t(u)/d$. In other words, we have designed an entirely deterministic protocol reminiscent of the recursions by which a random walk spreads a probability distribution. (See [16,17,79,107] for further applications as well as treatment of non-regular graphs, asynchronous protocols, failures of nodes e.t.c.) In Table 12, we give the rate at which the $\max_{v \in V} |\mu_v(t) - \eta|$ approaches 0 for various topologies on $n = 1M$ nodes.

5.4 Construction

In this section we turn our attention to construction and maintenance of well connected P2P topologies. Following the spirit of the previous sections, as well as the work of [62,92,114,125], we translate good connectivity to good expansion, conductance, and separation of λ_2 from 1. We further translate the fact that the construction concerns a P2P network to the following conditions. Peers arrive and leave the network dynamically, the algorithms must be strongly or weakly decentralized (in strong decentralization we mean that there is no central server, while in weak decentralization there is a constant number of central servers, however, the computational resources of each central server are of the same order of magnitude as those of an average peer) and finally, we wish to achieve low network overhead, in terms of messages, per addition or deletion. In the rest of section we deal only

Table 12: Convergence of maximum absolute error of an averaging gossip protocol.

Steps	$\lambda_2 = 0.7504$	$\lambda_2 = 0.8208$	$\lambda_2 = 0.9552$	$\lambda_2 = 0.9145$
0	$9.9 \cdot 10^5$	$9.9 \cdot 10^5$	$9.9 \cdot 10^5$	$9.9 \cdot 10^5$
10	$3.3 \cdot 10^3$	$4.4 \cdot 10^3$	$2.5 \cdot 10^3$	$2.0 \cdot 10^3$
20	7.45	145	51.6	30.9
30	2.36	7.20	6.05	1.86
40	$8.67 \cdot 10^{-2}$	0.540	3.63	0.651
50	$3.48 \cdot 10^{-3}$	$4.62 \cdot 10^{-2}$	2.29	0.266
60	$1.47 \cdot 10^{-4}$	$4.14 \cdot 10^{-3}$	1.45	0.109
70	$6.49 \cdot 10^{-6}$	$3.77 \cdot 10^{-4}$	0.918	$4.44 \cdot 10^{-2}$
80	$2.94 \cdot 10^{-7}$	$3.48 \cdot 10^{-5}$	0.580	$1.81 \cdot 10^{-2}$
90	$1.37 \cdot 10^{-8}$	$3.23 \cdot 10^{-6}$	0.367	$7.41 \cdot 10^{-3}$

Note: All topologies have $n = 1$M nodes. We used $A = 10^6$. All processes started with a value $A = 10^6$ on one node and the value 0 on all other nodes. The averaging gossip protocol converges to the value $\eta = 1$ on all nodes. We indicate the worst case error $\max_{v \in V} |\mu_t(v) - 1|$ for various values of t. Realize that when a node uses the estimate $\tilde{n} \simeq A/\mu_v(t)$ for the size n, then the absolute error $|\tilde{n} - n|$ becomes $10^6 \max_{v \in V} |\mu_t(v) - 1|$. Thus the above table suggests that, even for a strongly clustered topology with $\lambda_2 = 0.9552$, the estimate $\tilde{n}$ is $10^6 \pm 10^3$ for $n = 10^6$ after $t = 90$ simulation steps. For a less clustered topology with $\lambda_2 = 0.7504$ the same error is achieved after $t = 40$ simulation steps, while the error becomes truly 0 after $t = 90$ simulation steps.

with additions. Deletions can be handled as in [92].

In paragraph 5.4.1 we shall revisit the known randomized algorithms for constructing sparse graphs with good expansion properties. We call these baseline constructions. All known baseline algorithms construct an expander essentially by choosing the edges incident to a vertex uniformly at random, and independently for each vertex. This facilitates the probabilistic arguments that are subsequently used to establish expansion. We review the baseline probabilistic argument in Theorem 2. Thus, when constructing an expander on n vertices, a baseline construction uses $O(\log n)$ random bits per edge. In a distributed setting, when vertices arrive dynamically and a new vertex needs to extend an edge to an existing vertex, one may use $O(\log n)$ steps of a random walk on the existing graph to find a random existing vertex, thus simulating the baseline construction with $O(\log n)$ message overhead on the network. This is the approach of [92]. (Pandurangan et al. use the fact that deletions happen in a random way, and they use the "randomness" of deletions to connect new vertices [125]).

In Theorem 3 of paragraph 5.4.2 we show, analytically, that a certain baseline construction

achieves non-trivial expansion properties using constant number of random bits per new edge. When translated to overhead in network resources, this gives a heuristic to construct an expander with constant message overhead per new vertex. In particular, instead of taking $O(\log n)$ steps of a random walk per newly arriving vertex, we do the following. We keep a constant number of daemons which continuously simulate a random walk on the existing network and use the vertices visited by the daemons every c steps as sample points, where c is a constant. We stress that we do not wait $O(\log n)$ steps until the daemon randomizes. In paragraph 5.4.3 we report that, in experiment, the method achieves constant separation of λ_2 from 1 for c as small as 1 (sampling consecutive vertices visited by the daemons). The eigenvalue gap depends on c and on the average degree of the constructed graph, but does not depend on the size of the constructed graph, for n as large as 5M vertices. Note that the current size of Kazaa is thought to be 2M to 4M. (Performing experiments for more than 5M vertices was stressing the memory limitations of our machines.)

5.4.1 Baseline Construction of Expander Graphs

A_{BASE} is the following construction: On input n, the number of vertices, and d, the degree of every vertex, each vertex, independently, picks d vertices independently and uniformly at random among the set of all vertices, and connects with an (undirected) edge to each one of these vertices. Thus the total number of edges is nd and the expected degree of a vertex is $2d$. (It can be easily seen, using (24), that all vertices will have degree at most $O(\log n)$, almost surely.)

When we insist on all vertices having the same degree, then we simulate A_{BASE} by picking random perfect matchings, or random Hamilton cycles. In particular, $A_{\mathcal{M}}$ is the following construction: Pick d perfect matchings on n vertices independently and uniformly at random among all perfect matchings (assume that n is even), and consider the union of these perfect matchings. Finally, $A_{\mathcal{H}}$ is the following construction: Pick d Hamilton cycles on n vertices independently and uniformly at random among all Hamilton cycles, and consider the union of these Hamilton cycles.

Theorem 2 (Folklore.)**.** *Let $G(V, E)$ be the graph constructed by A_{BASE}. $G(V, E)$ is an expander, with high probability. In particular, there is a positive constant $\alpha < 1$ such that*

$$\Pr[\min_{S \subset V, |S| \leq |V|/2} \frac{|C(S)|}{|S|} \geq \alpha] \geq 1 - o(1) \ . \tag{32}$$

Proof. For a positive constant α, we say that a set of vertices S with $|S| \leq |V|/2$ is Bad if and only if $|C(S)| \leq \alpha|S|$. We will show that there exists a positive constant α such that

$$\Pr[\exists \text{ Bad } S] \ \leq \ o(1) \ . \tag{33}$$

The left hand side of (33) is

$$\sum_{k=1}^{n/2} \Pr[\exists \text{ Bad } S, \ |S| = k] \ . \tag{34}$$

Let us fix k in the above range. There are at most $\binom{n}{k}$ sets of vertices of cardinality k. We hence need to bound

$$\sum_{k=1}^{n/2} \binom{n}{k} \Pr[\text{a fixed set } S, |S| = k, \text{ is Bad}] \ . \tag{35}$$

We may now assume that the set S is fixed. Let $B \subset S$ be the set of vertices in S that chose to connect to vertices in $\bar{S}$. In order for S to be Bad, the cardinality $|B|$ must be at most αk. For each cardinality in the range 0 to αk, there are at most $\binom{k}{\alpha k}$ possibilities for $|S|$, and at most αk possibilities for the cardinality of B. We may now assume that the set B is also fixed. Finally, for fixed S and B, the probability that an edge picked by a vertex in $S \setminus B$ connects to a vertex in S is at most k/n, and there are $d(k-\alpha k)$ such edges. We may now write

$$\Pr[\text{fixed } S, |S| = k, \text{ is Bad}] \leq \alpha k \binom{k}{\alpha k} \left(\frac{k}{n}\right)^{dk(1-\alpha)} \ . \tag{36}$$

Combining (34), (35) and (36), (33) gives:

$$\begin{aligned} \Pr[\exists \text{ Bad } S] \ &\leq \ \textstyle\sum_{k=1}^{n/2} \Pr[\exists \text{ Bad} S, \ |S| = k] \\ &\leq \ \textstyle\sum_{k=1}^{n/2} \binom{n}{k} \alpha k \binom{k}{\alpha k} \left(\frac{k}{n}\right)^{dk(1-\alpha)} \\ &\leq \ \textstyle\sum_{k=1}^{n/2} \left(\frac{en}{k}\right)^{k} \alpha k \left(\frac{ek}{\alpha k}\right)^{\alpha k} \left(\frac{k}{n}\right)^{dk(1-\alpha)}, \text{ using } \left(\frac{n}{k}\right)^{k} \leq \binom{n}{k} \leq \left(\frac{en}{k}\right)^{k} \\ &\leq \ \textstyle\sum_{k=1}^{n/2} e'^{k} \left(\frac{k}{n}\right)^{k(d(1-\alpha)-1)} \end{aligned} \tag{37}$$

where $e' = e'(\alpha)$ is a constant. Now the last line of (37) is bounded by $o(1)$ if every term is bounded by $o(n^{-1})$, which is true for any $d \geq 2(1-\alpha)^{-1}$, since $\alpha < 1$. □

5.4.2 Baseline Construction of Expanders with Constant Overhead in Random Bits

Procedure A_{BASE} assumes that when a vertex chooses d other vertices to attach, these choices are independent and uniformly distributed in the integers 1 to n. Now consider the following pseudo-random number generator: A constant degree expander graph $\mathcal{H}$, with second eigenvalue μ_2, is

imposed over a set of n points labeled with the numbers 1 to n. We start a random walk on $\mathcal{H}$ from a point chosen uniformly at random, and, whenever the algorithm A_{BASE} needs a random point, we feed it with the current point of the random walk on $\mathcal{H}$. We call this algorithm A'_{BASE}. Realize that the random choices of A'_{BASE} are highly correlated, since they resulted from consecutive vertices visited by the random walk on $\mathcal{H}$. Nevertheless, we are able to establish a non-trivial expansion property for A_{BASE}, which essentially describes constant expansion of relatively large subsets of vertices. Our proof follows by the same probabilistic argument as Theorem 2, except we use (29) to bound the probability of correlated bad events:

Theorem 3. *Let $G(V,E)$ be a graph constructed by A'_{BASE}. There are positive constants α and β, $0<\beta<.5$, such that any subset S of at least $\beta|V|$ and at most $|V|/2$ vertices has cutset expansion α, almost surely. In particular,*

$$\Pr[\min_{S\subset V,\beta|V|\geq|S|\leq n/2}\frac{|C(S)|}{|S|}\geq\alpha]\geq 1-o(1)\ . \tag{38}$$

Proof. The reasoning is identical to the proof of Theorem 2, up to (36). At this point we use (29): What is the probability that $dk(1-\alpha)$ subsequent points of the random walk on $\mathcal{H}$ corresponded to ending up inside S, while the probability of falling outside S is at least $(n-k)/n\geq 1/2$? We apply (29) with $\epsilon=1$ and $p=1/2$ and get

$$\Pr[\text{fixed } S,|S|=k,\text{ is }\mathsf{Bad}]\leq\alpha k\binom{k}{\alpha k}8e^{-\frac{dk(1-\alpha)}{80}(1-\mu_2)}\ . \tag{39}$$

Now the final calculations become

$$\begin{aligned}\Pr[\exists\ \mathsf{Bad}S] &\leq \textstyle\sum_{k=\beta n}^{n/2}\binom{n}{k}\alpha k\binom{k}{\alpha k}8e^{-\frac{dk(1-\alpha)}{80}(1-\mu_2)}\\ &\leq \textstyle\sum_{k=\beta n}^{n/2}\left(\frac{en}{k}\right)^k\alpha k\left(\frac{ek}{\alpha k}\right)^{\alpha k}8e^{-\frac{dk(1-\alpha)}{80}(1-\mu_2)}\\ &\leq \textstyle\sum_{k=\beta n}^{n/2}e^{-\prime k}e^{\frac{dk(1-\alpha)}{80}(1-\mu_2)}\end{aligned} \tag{40}$$

where $e'=e'(\alpha,\beta)$ is a constant, and the last line of (40) is bounded by $o(1)$ for some constant $d=d(\alpha,\beta)$. Note that $k\geq\beta n$ is crucial to bound $(n/k)^k$ by e^k. □

5.4.3 Distributed Construction of Expanders with Constant Overhead on Network Resources

In this paragraph, we study how the concept of Paragraph 5.4.2 can be used to speed up the approach of [92]. We examine two algorithms.

Table 13: λ_2 of $A'_{\mathcal{H}}$ as a function of size and number of random walk steps.

Size (K)	Random	$c = \log_2 n$	c=5	c=3	c=1	c=0
10	0.7455	0.7445	0.7452	0.7506	0.7906	0.9999
50	0.7453	0.7457	0.7459	0.7508	0.7924	0.9999
100	0.7452	0.7453	0.7460	0.7504	0.7938	0.9999
500	0.7453	0.7453	0.7463	0.7504	0.7944	0.9999
1000	0.7453	0.7452	0.7462	0.7503	0.7956	0.9999
5000	0.7453	0.7454	0.7462	0.7504	0.8023	0.9999

5.4.3.1 $A'_{\mathcal{H}}$

This is an extension of the scheme proposed in [92]. The authors propose a scheme to implement the $A_{\mathcal{H}}$ construction in a distributed, decentralized environment. To ensure random placement of each arriving node in each of the d cycles, they propose techniques to estimate the size of the network n and then perform d random walks of length $O(\log n)$, thus their overhead is $O(\log n)$. Instead, we keep d daemons, one for each Hamilton cycle. These daemons move freely in the topology. When a new node arrives, it contacts the daemon associated with the i-th Hamilton cycle, for $1 \leq i \leq d$, and inserts itself between the peer that currently hosts daemon i and one of its two neighbors in cycle i. We require that the daemons perform c number of steps before allowing a new peer to join the topology. Observe that in [92] c is $O(\log n)$. In fact, when c is $O(\log n)$, the daemons can be allowed to start from any node of the topology, and this makes the algorithm of [92] fully decentralized. We measure the quality of the constructed topology by the second eigenvalue of the corresponding transition matrix. See Table 13 and Figure 23. It is obvious that λ_2 remains constant as the topology scaled from 10K to 5M nodes. On the other hand, λ_2 depended on c. However, notice that for $c = 1$, λ_2 is larger only by 0.05 compared to $c = \log n$. (As a sanity test, when c is 0, that is without randomization, there is no expansion.) λ_2 also depends on d. In Table 13 we give the case of $d = 3$ corresponding to degree 6. The trends are identical for larger values of d.

5.4.3.2 $A'_{\mathcal{M}}$

The existence of Hamilton cycles in $A_{\mathcal{H}}$ and $A'_{\mathcal{H}}$ is a good property since it guarantees connectivity. Maintaining Hamilton cycles however is difficult to implement. In this paragraph, we consider a

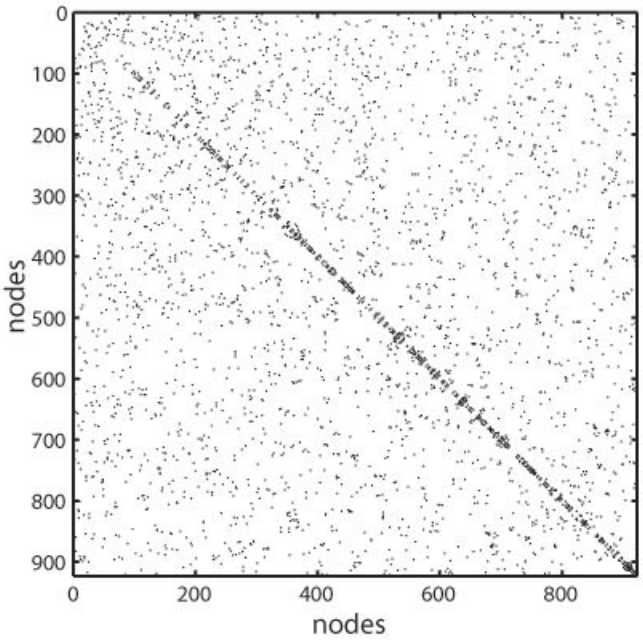


Figure 23: The connectivity matrix of a topology constructed using $A'_{\mathcal{H}}$ for $c = 1$. The strong dependencies are reflected in the concentration along the diagonal. However, there are many points away from the diagonal and the picture appears random.

Table 14: λ_2 of $A'_{\mathcal{M}}$ as a function of size, degree d and number of random walk steps c.

Size (K)	d=4 c=1	d=4 c=2	d=4 c=5	d=4 c=10	d=6 c=1	d=6 c=5	d=6 c=10
1	0.9754	0.8982	0.8711	0.8600	0.7782	0.7385	0.7392
10	0.9893	0.9131	0.8732	0.8654	0.7854	0.7468	0.7443
50	0.9939	0.9144	0.8777	0.8670	0.8015	0.7471	0.7450
100	0.9929	0.9312	0.8925	0.8673	0.8273	0.7470	0.7456
500	0.9969	0.9482	0.8833	0.8679	0.8332	0.7472	0.7454
1000	0.9995	0.9421	0.8861	0.8679	0.8287	0.7476	0.7455
5000	0.9996	0.9504	0.8846	0.8677	0.8348	0.7473	0.7454

distributed implementation of the $A_{\mathcal{M}}$ algorithm which does not require the existence of a special structure in the topology and thus is easier to implement. On the other hand, the price paid is an increased second eigenvalue. Also, the second eigenvalue does not remain relatively constant with the size of the network, but it increases slightly as we increase the number of peers. Our algorithm uses d daemons and works as follows. We model the arrival of a new node, as the arrival of two nodes X and Y, each with degree d. Upon the arrival, X and Y contact the central server to discover the location of the d daemons. Assume that daemon i is located at node A, and that the i-th neighbor of A is B. The connection between A and B is teared down and the new i-th neighbor of A is X and the new i-th neighbor of B is Y. Between each arrival, the daemons move c steps. The performance of the topology constructed by our algorithm, measured in terms of λ_2, for various sizes of the topology, values of d, and c are given in Table 14. Observe that the second eigenvalue is now a function of the degree, the number of steps and the size of the topology. However, for degree 6 and for $c = 1$, the second eigenvalue for $5M$ nodes is 0.8348 which is comparable to the corresponding entrance value in Table 13, which is 0.8023. The interpretation is that for current sizes of the network and for degree at least 6, both methods achieve equally good results.

5.4.3.3 Implementational issues

A final remark is due for the implemenational details of both $A'_{\mathcal{H}}$ and $A'_{\mathcal{M}}$. Despite the fact that in this work we propose primitives for construction and do not describe in any detail the design of the system, it is valid to question whether the proposed primitives can be implemented in a distributed system. The first question to ask is whether the random walker can handle the high churn rate of

typical peer-to-peer networks. Recall that the random walker must make a (small) number of steps before allowing new nodes to connect. We can address this question by maintaining many random walkers and assuming that new nodes or nodes that need more neighbors connect to a random small subset of them. The extra random walkers can only improve the randomization. The further practical issues related to the number of random walkers that need to exist to satisfy the demand and the way of creating, maintaining, and finding the random walkers are interesting questions to investigate, especially if they include real data.

Another question is the difficulty of implementing the proposed primitives in a distributed environment. The $A'_{\mathcal{H}}$ algorithm requires that all nodes must be arranged in Hamilton cycles and, thus, a global invariant must always hold. Maintaining global invariants is difficult in practice. On the other hand, the $A'_{\mathcal{M}}$ algorithm does not need to maintain any global invariant and should much easier to implement. Indeed, in practice, one may connect a newly arriving node to the positions of the d daemons without maintaining any invariant as Theorems 2 and 3 do not assume any invariants (however, establish good connectivity only for large sets). The reason that the simulation of $A'_{\mathcal{H}}$ (and $A'_{\mathcal{M}}$) is interesting is because this is precisely the case for which [92] establish analytically that an expander can be maintained with $O(\log n)$ overhead per new node. Our simulations suggest that constant overhead might be sufficient. (We have obtained simulation results similar to Tables 13 and 14 when a newly arriving nodes connects to d daemons without maintaining any invariant.)

CHAPTER VI

HYBRID SEARCHING SCHEMES

6.1 Introduction

Flooding is the predominant search technique in unstructured peer-to-peer (P2P) networks. If we measure performance as the number of exchanged messages per distinct response, flooding with small time-to-live performs well in regular networks. However, its performance deteriorates as the time-to-live increases, or if the topology of the underlying network is not regular [63]. In addition, flooding has poor granularity [105, 140].

Simulating a random walk has been proposed as an alternative search technique. In regular topologies, the performance of the random walk simulation method appears to be better than the performance of flooding. In addition, the random walk simulation method scales well and has excellent granularity. However, the simulation of a random walk is inherently sequential, which causes a large increase in the response time [63, 105].

We consider hybrid schemes which can be viewed as a random walk of substantially shorter length (and hence smaller response time), combined with very shallow floodings on every step of the random walk (see Figure 1d); similar hybrid schemes have been discussed explicitly in [2, 115] and implicitly in [32]. We shall refer to these schemes as random walks with lookahead. Alternatively, very shallow floodings, say of depth 1, can be thought of as a 1-step replication strategy, that is, where each node keeps a copy of the indices of his neighbors. In sparse networks, replication causes low network overhead, and benefits all future searches.

What is the analytic justification for the good performance of such hybrid schemes? Does the analysis suggest further efficient search algorithms? Is there a general abstraction, of which flooding and random walks are special instances? Can such an abstraction be useful in obtaining even more efficient search algorithms?

The first contribution of this chapter is to give analytic justification of why the simulation of a short random walk with shallow floodings on every step performs particularly well. The idea is the

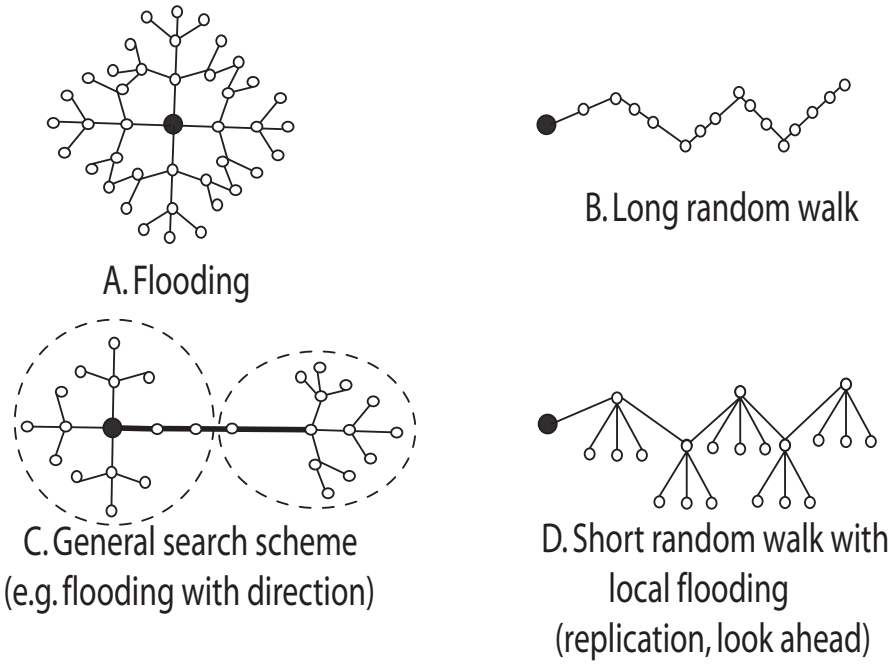


Figure 24: Categories of searching algorithms. (A) represents search by flooding. Flooding has good performance for small values of time-to-live. (B) represents search by random walk. The response time is proportional to the length of the walk. (C) represents a general search scheme, which is flooding amplified toward a critical direction. This is suitable in the case of clustered topologies, where the critical direction leads flooding outside a cluster. (D) represents a shorter random walk with local floodings. This decreases the response time and is particularly suitable when combined with 1-step replication.

following. Naturally, we expect that the time to discover a certain number of nodes using a random walk with shallow floodings will be somewhat smaller than in the simulation of a random walk without local floodings; the reason is that in each step of the random walk with shallow floodings we visit a node and all its neighbors, In particular, in a sparse network (say, with constant average degree and hence constant average gain per node), we would intuitively expect a constant saving in the response time. We show that there are sparse networks where the saving in the response time can be much sharper. In particular, we show that, in a standard random graph model, if the network has $\Theta(n)$ nodes of constant degree and $\Theta(\sqrt{n})$ nodes of degree $\Theta(\sqrt{n})$ then, the expected time to discover $\Omega(n)$ nodes is $O(\sqrt{n})$; this is in Theorem 8. The proof indicates that the reason for the dramatic improvement in the response time of random walk with local floodings is because the random walk biases the sampling towards the nodes that have high degree and hence yield more information using the shallow floodings. *From the practical point of view, Theorem 8 suggests that search by random walk with lookahead 1, or with 1-step replication, substantially amplifies its benefits when there is discrepancy on the degrees of the underlying topology.* (A remark is due about the assumption of discrepancy of $\Theta(\sqrt{n})$ in the degrees of a P2P network. Is this assumption realistic? Firstly, note that we show our results for large degrees $\beta\sqrt{n}$, where β can be a small constant; one may interpret the degrees of ultra-peers in current P2P networks for some $\beta = 0.1$. Secondly, we mainly use the $\Theta(\sqrt{n})$ assumption in our analytical results, for which this choice makes the calculations and the principal underlying phenomena cleaner. We can obtain similar results for much smaller values of the large degrees, with much more detailed calculations.)

On the other hand, we noted that flooding has poor performance when there is discrepancy in the degrees of the underlying network. The second contribution of this chapter is to rectify the performance of flooding in the case of a sparse network with a few vertices of large degrees. We study normalized flooding, where a vertex of small degree forwards a query to all his neighbors, while a vertex of large degree forwards a query to a small subset of its neighbors chosen uniformly at random. In Theorem 6, we show that, in a random network with $\Theta(n)$ nodes of constant degree and $\Theta(\sqrt{n})$ nodes of degree $\Theta(\sqrt{n})$, normalized flooding achieves performance comparable to flooding in a regular graph. In Theorem 9, we further show that normalized flooding with 1-step replication achieves performance comparable to random walk with 1-step replication, further indicating that

the gaining in 1-step replication comes from the bias of large degrees, and further strengthening the suggestion to use a small number of supernodes.

The moral of Theorems 8 and 9 can be thought of as follows. These theorems tell us that, *using local information of the network, in this case the degrees, we can get global benefit, by biasing the sampling towards the vertices with a lot of information*. Is there other local information that can be useful in searching? In a long sequence of theory papers, information concerning "edge criticality" has given novel approximation algorithms for NP-complete problems. In particular, [94] show that there are labels that can be assigned to edges, so that edges across bad cuts of the graph get heavier weights, and doing region growing according to these labels finds sparse cuts [154]. In addition, very roughly, [83, 84] show that these labels can be approximated by (repeated computations of) congestion under shortest path routings. Reminiscent of these techniques, *we define edge criticality metrics that identify edges belonging to sparse cuts*. We note that these metrics can be computed by local statistics that the network keeps anyway.

The third contribution of this chapter is to *define a generalization of searching*, of which flooding, random walks and random walks with lookahead are special instances. This is particularly simple to implement. In particular, we assume that a node initiates a search by assigning a budget, which is an upper bound on the number of messages that will be exchanged during the search. The node then partitions the budget, and may forward different partition classes of the budget to different neighbors. We show that this scheme is *particularly useful when edge criticality is known*. See Figure 1c. Suppose that the underlying network is clustered. Then, the thick edges will be assigned larger criticality, thus shifting a substantial part of the initial budget outside a specific cluster, and essentially initiate a new flooding in a different cluster. We report experimentation, where this heuristic has very good performance.

In summary, we show that (a) the existence of super-nodes improves searching performance if combined with suitable defined protocols, and (b) preferential treatment of links according to their criticality can further improve protocol performance.

The rest of this chapter is organized as follows. In Section 6.2 we give the graph models that we use and some crucial structural properties that will be later used in the proofs. In Section 6.3 we review the good behavior of flooding in regular graphs (Theorem 4), and argue that this behavior

deteriorates in graphs with supernodes (Theorem 5) while normalization rectifies this deterioration (Theorem 6). In Section 6.4 we review the behavior of random walks in regular graphs (Theorem 7), and argue that 1-step replication in graphs with supernodes can substantially improve the performance of the random walk method (Theorem 8). We further show that similar savings can be achieved by normalized flooding and 1-step replication (Theorem 9), further indicating that the savings come from the use of supernodes. In Section 6.5 we describe the generalized search scheme. In Section 6.6 we report experimental evaluation.

6.2 Random Graph Models

In this section we introduce random regular graphs, which aim to capture the behavior of a typical regular topology, and random graphs with supernodes, which aim to capture the typical behavior of a sparse network with a small number of large nodes. We review the graph theoretic notion of expansion and relate to the performance of flooding (see expressions 43, 45, and 41).

For graphs with supernodes, we establish new structural facts which characterize the neighborhoods of nodes with large degrees (Lemmas 3, 4 and 5).

We use the configurational random graph model. This is a standard model in the theory of random graphs as well as networking. In particular, for $d_1 \geq d_2 \geq \ldots \geq d_n$ denoting the degrees of a graph on n nodes, we generate a random graph as follows. First consider $D{=}\sum_{i=1}^{n} d_i$ mini-vertices corresponding to nodes in the natural way: the first d_1 mini-vertices correspond to node 1, the next d_2 mini-vertices correspond to node 2, and so on. Then consider a random perfect matching over the D mini-vertices, and a graph on the original n vertices defined by adding one link from node i to node j for each edge of the perfect matching that was connecting a mini-vertex corresponding to i with a mini-vertex corresponding to j. Note that this is a multigraph with self loops. In this section we maintain multiple links and self loops for analytic convenience.

Let d be a constant. By *random regular graph*, denoted $G_{n,d}$, we mean a random graph in the configurational model, with $d_i{=}d$, $1{\leq}i{\leq}n$. We next introduce a random graph model for graphs with supernodes. Let α and β be constants. Consider $\alpha n^{\frac{1}{2}}$ nodes of degree $\beta n^{\frac{1}{2}}$, called *large vertices*, and all the remaining nodes of degree d, called *small vertices*. By *random graph with supernodes*, denoted $G_{n,d,\alpha,\beta}$, we mean a random graph in the configurational model following the

above degree sequence. Note that random regular graphs and random graphs with supernodes have sum of degrees $D=dn$ and $D\simeq(\alpha\beta+d)n$ respectively, hence they are sparse, in the sense that the sum of the degrees of their vertices is $\Theta(n)$. Throughout this work, $\simeq$ means $1\pm o(1)$.

We further review the notion of vertex neighborhood and relate it to the performance of flooding. All the theorems in Section 6.3 are based on the characterization of vertex neighborhoods. We need the following definitions. Let $G(V,E)$ be an undirected graph, with $|V|=n$. Let S be a subset of vertices, $S\subset V$, and let $\bar{S}$ be its complement, $\bar{S}=V\setminus S$. Define the *vertex neighborhood* of S as $\Gamma(S)=\{u\in\bar{S}\ :\ (v,u)\in E\,,\text{ for some } v\in S\}$. Now let $S_i(v)$ be the set of vertices visited by flooding that initiated at vertex v with time-to-live i, and note that $S_i(v)=S_{i-1}(v)\cup\Gamma(S_{i-1}(v))$. Suppose that G is a d-regular graph. How many messages did each vertex propagate? The vertex v propagated d messages, and each vertex in $S_{i-1}(v)$ propagated at most $d-1$ messages, namely to all its neighbors except the one from which he received the query. Vertices in $\Gamma(S_{i-1}(v))$ were reached with time-to-live 0 and hence did not propagate any messages. Now we may upper bound the ratio of distinct responses over number of messages

$$\frac{|S_{i-1}|+|\Gamma(S_{i-1}(v))|}{(d-1)|S_{i-1}(v)|}=\frac{1}{d-1}\left(1+\frac{|\Gamma(S_{i-1}(v))|}{|S_{i-1}(v)|}\right)\,. \tag{41}$$

Since, 41 describes the performance of searching, it is important to examine the ratio $\Gamma(S_{i-1}(v))/|S_{i-1}(v)|$.

For a random regular graph $G_{n,d}$, [21] show the following property. For an arbitrary subset of vertices S of a graph G, define the *cutset* of S, $\nabla(S)$, as $\nabla(S)=\{(v,u)\in E\ :\ v\in S, u\in\bar{S}\}$. Let $S_i(v)$ be the set of vertices reached by flooding with time-to-live i, as above. [21] show that, with probability $1-o(n^{-2})$, for all vertices v and for all $i:(d-1)^i\le n^{\frac{1}{2}}\log n$,

$$|\nabla(S_i(v))|\geq\simeq(d-1)^i \tag{42}$$

We claim that this implies that, for all vertices v, and for all $i:(d-1)^i\le n^{\frac{1}{2}}$,

$$|\Gamma(S_i(v))|\geq\simeq(d-1)^i \tag{43}$$

To see this, note that (42) applied to $S_{i+1}(v)$ gives $|\nabla(S_{i+1}(v))|\geq\simeq(d-1)^{i+1}$. But all edges in $\nabla(S_{i+1}(v))$ must be incident to a vertex in $\Gamma(S_i(v))$, and each vertex in $\Gamma(S_i(v))$ can yield at most $d-1$ edges in $\nabla(S_{i+1}(v))$, which implies that $|\Gamma(S_i(v))|\geq\simeq(d-1)^i$.

For the random graph $G_{n,d}$, [57] further claim:

$$|\nabla(S)| \geq \begin{cases} (1 - O(\frac{1}{\sqrt{d}} + \epsilon))d|S| & |S| \leq \epsilon|V| \\ d|S|/4 & \epsilon|V| \leq |S| \leq |V|/2 \end{cases} \tag{44}$$

which immediately implies

$$|\Gamma(S)| \geq \begin{cases} (1 - O(\frac{1}{\sqrt{d}} + \epsilon))|S| & |S| \leq \epsilon|V| \\ |S|/4 & \epsilon|V| \leq |S| \leq |V|/2 \end{cases} \tag{45}$$

We will use (43) and (45) in the characterization of flooding in Section 6.3.

We proceed to discuss structural properties for graphs with supernodes. The crucial structural properties are that each small node is incident to a large degree node with constant probability, and each large node has, in expectation and with sharp concentration, a constant fraction of its edges incident to distinct large nodes and a constant fraction of its edges incident to small degree nodes. We will also use the following form of Chernoff bounds [145]. Let X_i, $i = 1, \ldots, N$, be independent random variables with $\Pr[X_i = 1] = p_i$ and $\Pr[X_i = 0] = 1 - p_i$. Let $X = \sum_{i=1}^{N} X_i$ and let $p = (\sum_{i=1} Np_i)/n$. Then,

$$\Pr[X - pN < -\Delta] < e^{-\frac{\Delta^2}{2pN}} \tag{46}$$

and

$$\Pr[X - pN > \Delta] < e^{\frac{\Delta^2}{2pN} + \frac{\Delta^3}{2(pN)^3}} \quad . \tag{47}$$

More formally, the structural facts for graphs with supernodes are Lemmas 3, 4 and 5 below.

Lemma 3. *Let $G = G_{n,d,\alpha,\beta}$ be a random graph with supernodes, and let ϵ be any constant $\epsilon < \max\{\alpha, \beta\}$. Then, with all but exponentially vanishing probability, every large vertex of G has $\frac{(\alpha-\epsilon)\beta}{d+\alpha\beta} \frac{\epsilon n^{\frac{1}{2}}}{2}$ distinct large neighbors.*

Proof. Let v be a large vertex. Suppose that $N = \epsilon n^{\frac{1}{2}}$ distinct neighbors of v are known to be distinct large vertices. What is the probability that v is incident to an additional distinct large vertex? There are $(\alpha - \epsilon)n^{\frac{1}{2}}$ remaining large vertices, hence the remaining total degree on large vertices is $(\alpha - \epsilon)n^{\frac{1}{2}}\beta n^{\frac{1}{2}}$. The total degree of all vertices is $(d + \alpha\beta)n$. Hence the probability that v sees an additional distinct large degree vertex, given that it has seen $\epsilon n^{\frac{1}{2}}$ distinct vertices is at least

$p = \frac{(\alpha-\epsilon)\beta}{d+\alpha\beta}$. We may now bound the probability that v is incident to less than $pN/2$ distinct vertices by observing that this probability is smaller than the probability in N independent experiments, each with probability of success p, there were less than $\Delta = pN/2$ successes. Using (46) above, we get exponentially small tails. □

Lemma 4. *Let $G = G_{n,d,\alpha,\beta}$ be a random graph with supernodes. Then, with all but exponentially vanishing probability, every large vertex of G has $\frac{d}{d+\alpha\beta}\frac{\beta n^{\frac{1}{2}}}{2}$ edges incident to (not necessarily distinct) small neighbors.*

Proof. Let v be a large vertex. Let $N = \beta n^{\frac{1}{2}}$. Suppose that $N-1$ edges incident to v are known to have their other endpoint incident to a small vertex. Then, the probability that the last edge is also incident to a small vertex is $\frac{dn-d}{(d+\alpha\beta)n} \simeq \frac{d}{d+\alpha\beta} = p$. We may now bound the probability that v has less than $\frac{d}{d+\alpha\beta}\frac{\beta n^{\frac{1}{2}}}{2}$ edges incident to small vertices by the probability that in N independent experiments, each with probability of success p, there were fewer than $\Delta = pN/2$ successes. By (46), this is exponentially small. □

Lemma 5. *Let $G = G_{n,d,\alpha,\beta}$ be a random graph with supernodes. Then, with all but exponentially vanishing probability, every large vertex v has $\Gamma(\{v\}) \cup \Gamma\left(\Gamma(\{v\})\right) \geq \frac{(\alpha-\epsilon)\epsilon\beta^2}{4(d+\alpha\beta)^2} n$.*

Proof. By Lemma 3, v has $\frac{\alpha-\epsilon}{d+\alpha\beta}\frac{\epsilon\beta n^{\frac{1}{2}}}{2}$ distinct large neighbors. By Lemma 4, each distinct large neighbor has $\frac{d}{d+\alpha\beta}\frac{\beta n^{\frac{1}{2}}}{2}$ distinct edges incident to small vertices. But each small vertex has at most d incident edges, and the statement of the claim follows. □

6.3 Flooding and Normalization

Flooding is the predominant search technique in unstructured peer-to-peer networks. Such floodings are typically parameterized by a time-to-live, τ. In particular, a node initiates a search by propagating a request, together with a time-to-live τ, to all his neighbors. Without loss of generality, we may think of the request as an exploration of the network: "if you get this message for the first time, then report your presence (e.g. address) to the initiator of the request". Flooding proceeds as follows. The first time that a node receives a request with time-to-live t, the node responds to the request

and, if $t > 0$, the node propagates the same request to all his neighbors. If a node receives the same request multiple times, then it will neither respond nor propagate it.

We quantify the performance of flooding by the *number of responses*, the *response time* (we assume that the delay of a particular response is proportional to the number of hops between the initiator of the query and the responding node), and by the *number of propagated messages*. Clearly, the number of responses and the response time quantify the quality of service perceived by the initiator of the search, and the number of propagated messages quantify the overhead perceived by the network. In practice, flooding is known to perform very well for small values of τ, however the performance does not scale well with τ. In addition flooding has poor *granularity*.

When a graph is not regular, then the performance of flooding deteriorates. In particular, when large degree vertices are reached, then these cause a sudden sharp increase in the number of neighbors they introduce (hence poor granularity), which, in turn, causes a lot of shared edges (hence poor performance in terms of number of messages per distinct number of discovered nodes). We therefore consider *normalized flooding* which is the following algorithm. Let $d_{\min}$ be the minimum degree of the network. In normalized flooding, when a node of degree $d_{\min}$ receives a query, the node propagates the query to all his neighbors (except the one which forwarded the query). When however a node of larger degree receives a query, the node propagates the query only to $d_{\min}$ of his neighbors, chosen uniformly at random from the entire set of his neighbors (except the one which forwarded the query). This is the natural normalization, and it is well known common practice (e.g. see [67]).

In Theorem 4 we establish the good behavior of flooding in regular graphs. The proof of this theorem is based on known structural properties of random regular graphs. Our contribution is to translate these properties in the context of flooding, as expressed in (41). It is important to notice that the upper bounds in Theorem 4 differentiate between ranges of the time-to-live, and clearly suggest that the guarantees on the performance of flooding deteriorate as the time-to-live increases. The number of distinct nodes discovered increases exponentially, and this indicates poor granularity.

Theorem 5 is a lower bound for flooding in graphs with supernodes. It indicates that, without normalization, a large vertex is discovered for a very small value of time-to-live, hence even poorer granularity. Theorem 6 indicates that normalized flooding in graphs with supernodes can rectify

the performance of flooding. In particular, it brings the performance of normalized flooding, up to order of magnitude, to the performance of flooding in regular graphs (we show this for small values of time-to-live where flooding in regular graphs has its best behavior). The proofs of Theorems 5 and 6 make critical use of the structural properties established in Lemmas 3, 4, and 5.

For analytic convenience in the analysis of normalized flooding, we think of the following finer structure in $G_{n,d,\alpha,\beta}$. Each set of $\beta n^{\frac{1}{2}}$ minivertices corresponding to a large vertex is further partitioned into minigroups of d minivertices. We may now think of $G_{n,d,\alpha,\beta}$ as a random regular graph with all minigroups corresponding to the same large vertex contracted to a single large vertex.

Theorem 4. *Let $G_{n,d}$ be a random regular graph, let v be a node of $G_{n,d}$, and consider a flooding in the basic scenario initiated by v with time-to-live τ. Let S be the number of distinct nodes queried by this flooding and suppose that $|S| \leq |V|/2$. Then, for $\tau \leq \frac{\log n}{2\log(d-1)}$, the number of distinct responses is $|S| \geq\simeq (d-1)^{\tau-1}$ and the number of distinct responses per message is at least $\simeq \frac{1}{d-1}\left(2-O(\frac{1}{\sqrt{d}})\right)$, almost surely. Furthermore, for any S with $|S| \leq \epsilon|V|$, $\epsilon < 1/2$, the number of distinct responses is $|S| \geq\simeq \left(2-O(\frac{1}{\sqrt{d}}+\epsilon)\right)^{\tau}$ and the number of distinct responses per message is at least $\simeq \frac{1}{d-1}\left(2-O(\frac{1}{\sqrt{d}})\right)$, almost surely. Finally, for any S with $|S| \leq |V|/2$, the number of distinct responses is $|S| \geq\simeq (1+\frac{1}{4})^{\tau}$ and the number of distinct responses per message is at least $\frac{1}{d-1}(1+\gamma)$, almost surely.*

Proof. For random regular graphs, the behavior of flooding for $\tau \leq \frac{\log n}{2\log(d-1)}$ is derived from the fact that breadth-first-search with bounded depth, in particular until $n^{\frac{1}{2}}\log n$ nodes are visited, has very good behavior, almost surely. We expressed this in formula (43). Now the performance claimed in Fact 4 for $\tau \leq \frac{\log n}{2(d-1)}$ can be obtained as follows. Using (43), the number of distinct responses received with time-to-live τ is

$$\begin{array}{rcl} \sum_{i=0}^{\tau-1} |\Gamma(S_i(v))| & \geq & \simeq \sum_{i=0}^{\tau-1}(d-1)^i \\ & = & \frac{(d-1)^{\tau+1}-1}{d-2} \\ & \geq & (d-1)^{\tau} \ . \end{array} \tag{48}$$

For the number of messages per distinct response we use (45). Now the number of messages per distinct response follows by substituting (45) in (41). In particular, since $|S| \leq (d-1)^{\tau}$, for $\tau \leq \frac{\log n}{2\log(d-1)}$ we get $|S| \leq |V|^{\frac{1}{2}}$, which implies $\epsilon \leq |V|^{-\frac{1}{2}}$, and hence $|\Gamma(S_{\tau-1}(v))|/|S_{\tau-1}(v)|$ is at least $1-O(1/\sqrt{d})$. □

Theorem 5. *Let $G_{n,d,\alpha,\beta}$ be a random graph with supernodes, let v be a node of $G_{n,d,\alpha,\beta}$ of degree d, and consider a flooding initiated by v in the basic flooding scenario. Then, for some time-to-live $\tau = \Theta(\log\log n)$, the number of distinct responses is $\Omega(n)$, almost surely.*

Proof. Consider flooding with time-to-live $\tau \simeq c\log_{d-1}\log n + 1$, for some constant c. We first argue that, with all but polynomially vanishing probability, a large vertex is found. To see this, consider the vertices visited with time-to-live up to $\tau-1$, and suppose that this set does not contain a large degree vertex. This set then can be thought of as the result of flooding on a random d-regular graph, and by (42), the cutset of this set has at least $(d-1)^{\tau-1}$ edges. The probability that the other endpoint of each edge in this cutset is a small vertex is $\frac{d}{d+\alpha\beta}$. Thus the probability that no vertex in $\Gamma(S_{\tau-1}(v))$ is large can be bounded by $(\frac{d}{d+\alpha\beta})^{(d-1)^{\tau-1}} = (\frac{d}{d+\alpha\beta})^{c\log n}$. So we know that, almost surely, within the first $O(\log\log n)$ steps we will see a large vertex. Now, by Lemma 5 this vertex will explore $\Omega(n)$ vertices in two more steps of the flooding. □

Theorem 6. *Let $G_{n,d,\alpha,\beta}$ be a random graph with supernodes, let v be a node of $G_{n,d,\alpha,\beta}$, and consider a normalized flooding initiated by v with time-to-live $\tau \leq \frac{\log n}{2\log(d-1)}$. Then, the number of distinct responses is $\Omega((d-1)^{\tau-1})$ and the number of messages per response is $O(1)$, almost surely.*

Proof. By Theorem 4, in τ, the number of minigroups seen is $(d-1)^{\tau-1}$. The expected number of small vertices is $\frac{d}{(d+\alpha\beta)}(d-1)^{\tau-1}$. Now using (46), the probability that less than $\frac{d}{2(d+\alpha\beta)}(d-1)^{\tau-1}$ are seen is vanishingly small. □

6.4 Random Walks and Replication

Everything else being equal, the best way to search a graph would be by uniform sampling. Assuming that a random node of the network could be generated efficiently, we could take k such samples simultaneously at cost one message per sample. By the well known coupon collection theorem (uniform sampling with replacement), for any $1 \leq k \leq n$, the expected number of samples to visit k distinct nodes is $(H_n - H_{n-k})n$, where H_i is the i-th harmonic number. In particular, the expected number of samples to visit all the nodes is $n\log n$ and, for any constant $\epsilon < 1$, the expected number

of samples to visit ϵn distinct nodes is $\frac{\epsilon}{1-\epsilon}n$. Thus, the amount of network overhead per distinct response can come arbitrarily close to 1, while retrieving a constant fraction of the search space. In addition, all the samples can be drawn simultaneously. The drawback of course is that it is not known how to implement uniform sampling in the relevant application context.

The *random walk* method has been proposed as a practical alternative to implement uniform sampling [63, 105]. In particular, in several random graph models, the so-called mixing time of the random walk, which is the number of simulation steps in order for the random walk to reach a distribution close (for sampling purposes) to uniform, is $O(\log n)$. This means that we may simulate k uniform samples with $O(\log n)$ random walk steps for each uniform sample. Since the random walks can be simulated in parallel, and assuming that the response delay of a random walk is proportional to the number of simulation steps of the walk, we get maximum response time $O(\log n)$, overhead at most $O(k \log n)$, while achieving performance similar to uniform sampling. The drawback of this approach is the network overhead which scales as $O(\log n)$. On the positive side, the theory of cover times [44] [43], complexity theory [61, 72] and extensive experimentation [63] suggest that this overhead can be reduced to a constant by taking $O(\log n)$ steps to randomize and then using k *successive steps* of the random walk in the place of independent samples. The drawback however is that the approach is inherently sequential and hence introduces maximum response time at least k.

The behavior of the random walk method for regular graphs is in Theorem 7 below. We give this well known theorem without proofs.

Theorem 7. *Let $G_{n,d}$ be a random regular graph, let v be a node of $G_{n,d}$, and consider a random walk starting at v. Then, for any k with $1 \leq k < n$, the expected number of messages and response time to get k distinct responses is at most $(H_n - H_{n-k})nO(\log n)$, almost surely. In addition, the expected number of messages to get n distinct responses is $\frac{d-1}{d-2}n \log n$, almost surely.*

One way to reduce the response time is to perform a much shorter walk, and in addition perform shallow floodings on each step of the walk. We call this method *random walk with lookahead*. In regular graphs, for constant lookahead (flooding with constant depth) we expect a constant saving in the response time. Here we observe that the savings in the response time are much sharper, if the

graph has supernodes; similar results have been also observed in [2] and [115] in power law graphs. In particular, Theorem 8 suggests that, for lookahead 1, we may visit $\Omega(n)$ nodes with response time $O(n^{\frac{1}{2}})$. The proof of Theorem 8 makes crucial use of the structural properties of graphs with supernodes that were established in Section 6.2.

Let us further consider 1-step replication. In this scenario, a node maintains information about all his neighbors and, when queried, includes this information in his response. In experiment, [105] observed very good performance of the sequential random walk method in a network with 1-step replication. In a sparse network, 1-step replication can be implemented with a one-time linear overhead where all edges exchange the information of their endpoints, while the benefit of this replication can be experienced by all future queries (see also [106] for another application of lookahead). Theorem 8 establishes the performance of 1-step replication. Realize that lookahead and 1-step replication are different implementations of the notion of short random walks with flooding with time-to-live 1 at every step.

Intuitively, the reason why lookahead and 1-step replication offer very sharp savings in graphs with supernodes is that the stationary distribution of the random walk has sharp bias towards large vertices, which yield a lot of information about their neighbors. Is random walk the only way to achieve such savings? In Theorem 9 we show that normalized flooding can achieve similar savings (up to order of magnitude). Intuitively, the reason is that normalized flooding can be also thought of as mimicking sampling from a distribution with sharp bias towards large vertices.

Theorem 8. *Consider any ϵ such that $0 \leq \epsilon < \frac{1}{2}$. Let $G_{n,d,\alpha,\beta}$ be a random graph with supernodes, let v be a node of $G_{n,d,\alpha,\beta}$, and consider a random walk starting at v. Then, in the 1-step replication scenario, the expected number of messages and response time to obtain $\frac{\alpha\beta n^{1-\epsilon}}{4d} = \Omega(n^{1-\epsilon})$ distinct responses is $\frac{d+\alpha\beta}{\alpha\beta} O(\log n) \frac{\alpha n^{\frac{1}{2}-\epsilon}}{2} = O(n^{\frac{1}{2}-\epsilon} \log n)$, almost surely.*

Proof. The stationary distribution of the random walk is as follows. Each large vertex has probability $\beta/(d+\alpha\beta)\sqrt{n}$, and each small vertex has probability $d/(d+\alpha\beta)n$. Since there are $\alpha\sqrt{n}$ large vertices, the stationary probability of all large vertices is $\frac{\alpha\beta}{d+\alpha\beta}$. Now for the simulation of the random walk, using the conductance result of [62], we get that, in $O(\log n)$ simulation steps we will

have a vertex sampled from a distribution which is arbitrarily close to the stationary. Hence, in expected $\frac{d+\alpha\beta}{\alpha\beta}O(\log n)$ simulation steps we get a large vertex, and, by coupon collection, in expected $\sum_{j=1}^{\alpha n^{\frac{1}{2}-\epsilon}/2} \frac{\alpha n^{\frac{1}{2}}}{\alpha n^{\frac{1}{2}}-j+1} = \frac{\alpha n^{\frac{1}{2}-\epsilon}}{2}$ large vertices we get $\frac{\alpha n^{\frac{1}{2}-\epsilon}}{2}$ distinct large vertices. So in expected $\frac{d+\alpha\beta}{\alpha\beta}O(\log n)\frac{\alpha n^{\frac{1}{2}-\epsilon}}{2}$ simulation steps we get $\frac{\alpha n^{\frac{1}{2}-\epsilon}}{2}$ distinct large vertices. Since each large vertex has $\frac{\beta n^{\frac{1}{2}}}{2}$ edges incident to small vertices, we get $\frac{\alpha\beta n^{1-\epsilon}}{4}$ edges incident to small vertices. But each small vertex can be incident to at most d large vertices, which completes the proof □

Theorem 9. *Let $G_{n,d,\alpha,\beta}$ be a random graph with supernodes, let v be a node of $G_{n,d,\alpha,\beta}$. Consider normalized flooding starting at v with time-to-live $\tau \simeq \frac{\log n}{2\log(d-1)}$. Then, in the 1-step replication scenario, the number of distinct responses is at least $\frac{(d-1)^{\tau-1}\alpha b^2 n^{\frac{1}{2}}}{8d(d+\alpha\beta)} = \Omega(n)$, almost surely, and the number of messages is at most $(2 - O(\frac{1}{\sqrt{d}})(d-1)^{\tau} = O(n^{\frac{1}{2}})$.*

Proof. By reasoning as in the proof of Theorem 4, there will be $(d-1)^{\tau-1}$ minigroups. Using (46), there will be $\frac{(d-1)^{\tau-1}\alpha\beta}{2(d+\alpha\beta)}$ minigroups corresponding to large vertices. How many minigroups corresponding to distinct large vertices were found? The probability that a group found corresponded to a distinct large vertex is at least $\frac{\alpha n^{\frac{1}{2}} - \frac{(d-1)^{\tau-1}\alpha\beta}{2(d+\alpha\beta)}}{\alpha n^{\frac{1}{2}}} \geq 1/2$. Using (46), there will be $\frac{(d-1)^{\tau-1}\alpha\beta}{4(d+\alpha\beta)}$ distinct large vertices. Now, using Lemma 4, each distinct large vertex has $\frac{\beta n^{\frac{1}{2}}}{2}$ incident small vertices, and since each small vertex can be incident to at most d large vertices, we get a total of at least $\frac{(d-1)^{\tau-1}\alpha b^2 n^{\frac{1}{2}}}{8d(d+\alpha\beta)}$ distinct small vertices. □

6.5 Generalized Search Schemes

We now describe a new searching scheme that allows very fine granularity of the number of messages that will be used for searching, like searching with random walks, and still allows very fast searching, like searching with flooding. A node initiates a search by picking a *budget* k, which is the number of messages that will propagate in the network. Assuming that the node has d neighbors, then the node distributes its budget by picking d integers $k_1, \ldots, k_d$, with $k_i \geq 0$, $1 \leq i \leq d$, and $k_1 + \ldots + k_d = k$. Then, it forwards the query to node i with budget equal to k_i (if $k_i = 0$ then the query is not forwarded to node i). Each neighbor i will reduce the budget received by 1 and repeat the same process if the new budget is greater than 0. Because the generalized searching is sensitive

to budgets, if a node receives the same query for a second time, from a different neighbor, then it will forwarded it again. Of course, the most critical task is the choice of k_1 to k_d.

The generalized searching scheme has both random walks and floodings as a special instance. To simulate random walk each node picks a neighbor i at random and assigns to it the remaining budget (say $k-1$); all other neighbors are assigned a budget of 0 and thus no query message is forwarded to them. Flooding in regular graphs can also be simulated easily. Each node divides the budget equally to all its neighbors minus the neighbor from which it received the query (if it did not initiate the message itself). To compute the initial budget, the node initiating the query needs to have an estimate of the number of messages that a regular flooding with TTL τ would generate; in regular graphs with degree d this can be $(d-1)^\tau$. In general graphs it is not possible to simulate flooding exactly. A good approximation however is to divide the budget to each neighbor according to their degrees.

The main advantage of the generalized searching is that it allows arbitrary assignment of budget to each neighbor. Intuitively, a node should assign a larger budget to neighbors through which more peers can be reached. This is particularly important for topologies that have clusters. Assume a topology with two clusters and few edges between the clusters. When the search reaches a border node, then that node needs to forward the query with larger budget to the other cluster since the nodes in its own cluster can be reached from different paths. Moreover, the query should propagate with higher weight towards the border nodes. How should a node allocate the budget to its neighbors to achieve that behavior? In other words, what are good heuristics to compute the k_i's?

Let us use the generic term "edge criticality" to denote metrics related to the importance of the edges. One way to define the criticality of an edge is as the number of shortest-hop paths between any pair of nodes of the network that use that edge. The few inter-cluster edges and the edges that lead to them are assigned higher criticality since a large number of paths will go through them. Thus, an effective way to perform searching in topologies with clusters is to divide the budget according to the criticality of the edge that connects to each neighbor.

Computing the edge criticality, according to the shortest-hop path definition, is extremely difficult since each node should perform a flooding to the entire network to discover the shortest-hop paths to each other node. A more practical approach is to have only a subset of nodes discover the

shortest hop paths to each node. Experimentally, we have observed that computing the shortest-paths from less than 2% of the nodes was sufficient. Another practical approach is to discover the shortest-hop paths to all other nodes that have a distance less than a specified value. In current P2P network, like Gnutella, each node periodically floods the network with TTL 7 to discover the peers that are in its horizon. If the flooding message contained the identity of the originating node, then intermediate nodes could monitor the queries and the replies to infer how many shortest-paths go through them. In general, estimating the edge criticality is possible in current networks, and, moreover, for the purpose of searching with budgets, a rough estimation of the edge criticality gives very good results as we shall see later in Section 6.6.5.

Note that current peer-to-peer networks already implement heuristics that use information collected locally to improve the performance of the network. In implementations of the gnutella protocol, including mutella [119], the nodes estimate the so-called "efficiency" of each link, which is the number of unique queries over the total number of queries received from that link. Nodes preferentially drop links with low efficiency. Thus, in current peer-to-peer networks, nodes use local measurements to improve the performance of the network. Our scheme fits in the same framework and uses locally collected measurements to improve the performance of searching.

There are many other ways to define edge criticality. In a long sequence of theory papers, information concerning "edge criticality" has given novel approximation algorithms for NP-complete problems. In particular, [94] show that there are labels that can be assigned to edges, so that edges across bad cuts of the graph get heavier weights, and doing region growing (weighted breadth first search) finds sparse cuts [154]. Furthermore, [83,84] show that these labels can be approximated by (repeated computations of) congestion under shortest path routings. Our notion of edge criticality is reminiscent of these techniques.

Another approach to define edge criticality is by considering the principal eigenvectors of the stochastic normalization of the connectivity matrix of the network topology [23, 37, 66, 80]. Some of these quantities can be also computed efficiently in a distributed setting [80]. It would be very interesting (also, rather hard) to obtain analytical results for heuristics such as the ones proposed in this section.

6.6 Experimental Evaluation

In this section, we study experimentally the performance of the hybrid and generalized searching schemes, and the benefits of 1-step replication and lookahead. The experimental results validate the analytical results of the previous sections, and quantify in more detail the performance of the proposed heuristics.

In Section 6.6.1 we present our experimental methodology. In Section 6.6.2 we compare normalized flooding to regular flooding. In Sections 6.6.3 and 6.6.4 we study 1-step replication and shallow lookaheads respectively. In Section 6.6.5 we study the generalized search scheme.

6.6.1 Methodology

6.6.1.1 Performance Metrics

In our experiments we are interested in characterizing the performance of searching. We choose some distinct random nodes and perform searching starting from these nodes with the algorithms described in Sections 6.3, 6.4, and 6.5 (typically we use 500 nodes). We measure the number of distinct peers visited per searching per node, which we call *hits*. Our definition of hits relates directly to the standard definition of the number of copies of a specific object discovered by searching, assuming that the copies of the requested information are placed at random in the network. We also measure the number of propagated messages, which directly relates to the load injected in the network for searching. In addition, we measure the response time of the searching, i.e. the maximum time it takes for the query to complete. More specifically we use the following metrics:
Median and Mean number of distinct peers discovered (hits). Searching algorithms should maximize the median and mean number of distinct peers. The median is a more robust metric, since, in topologies with large irregularities in the degrees, it is possible to measure relatively large mean values because few searches may reach a very large number of users and increase the mean value.
Minimum, Maximum, and Standard Deviation of the number of hits. A large minimum value is important in order to guarantee that the algorithm will have a good worst case performance. The range between the minimum and the maximum values relates to the variation of the performance of the algorithm. The variation is measured using the standard deviation. We believe that algorithms with larger minimum values and smaller variation are preferable.

Number of messages. Good searching schemes strive to minimize the number of messages used to discover as much information as possible. In order to perform a fair comparison of the different searching algorithms we require that they use the same number of messages. Since it is difficult to configure the parameters of each algorithm to guarantee the exact same number of messages, we require that the expected number of messages used in each experiment is approximately the same for all algorithms.

Granularity of number of messages. This is a qualitatively and not quantitative metric. In flooding based algorithms it is difficult to control the parameters of the algorithm, usually the time-to-live, to use a pre-specified number of messages for searching. Linear increases in the TTL result usually in exponential increases in the number of messages. Algorithms with finer granularity are preferable since the user can control the number of messages to reach an adequate number of users (hoping to find enough copies of an item), but, still, not flood the entire network.

Response time. We also measure the maximum running time of each algorithm. In this study we assume a very simple discrete-time model. Each node receives queries from its neighbors and at the same time processes them and forwards copies of the queries, if necessary, to its neighbors. The latter queries will be received at the next unit of time. For all our schemes it is easy to compute the running time of the algorithm, or an upper bound of it. For example, the searching time for flooding with TTL τ is τ. Despite the fact that we do not model many important parameters that affect the searching time, like for example propagation and queuing delays, we believe that our definition of running time can be used to judge the relative performance of the different algorithms.

6.6.1.2 Topologies

We are interested in studying the performance of the searching algorithms in networks with irregularities in the node degrees and in networks with clustered node topologies. Both cases are typical in complex and unstructured communication networks. In peer-to-peer networks, users with very good network connectivity may decide to serve as supernodes. Typically, these users have a much larger number of neighbors; in the Gnutella network for example the average user has 4-6 network connections, whereas a supernode may connect to 20-30 other peers and in many cases have even more neighbors. Also, a common pattern that appears in every unstructured communication network is

the clusterness of the topology. The existence of these clusters has been shown to greatly affect the performance of various communication functions in these networks, including searching [63, 66]. There are other parameters that may affect the performance of searching, including the dynamic nature of the topologies, that have been studied elsewhere and which we ignore in this study.

We will use the following synthetic topologies to compare the searching algorithms. Currently, there are limited real data for operational peer-to-peer networks mainly due to the difficulty in collecting such topological information.

Random d-regular Graphs. Extensive analytical work has shown that random d-regular graphs have good properties, including low diameter, good connectivity (i.e. there are no clusters of nodes), small second eigenvalue, and good conductance. We will use d-regular random graphs as a canonical model of a well connected network. Topologies generated with that model will serve as baseline for comparisons. Moreover, the topologies of third generation peer-to-peer networks, like BitTorrent, that use a centralized server to control how the topology is formed, may be more accurately modeled by well connected topologies.

Power Law Graphs. Many seemingly complex networks, including the Internet at the AS level, the Web Graph, and many others, have been shown to be characterized by power-laws. Most typically the degrees of the nodes of the network follow a power-law. The power-law is usually characterized by a parameter λ called the powerlaw exponent, which relates to the probability of observing nodes of high degree according to the formula $\Pr[\text{degree} > x] \sim 1/x^{\lambda}$. We use topologies with $\lambda = 1.4, 2.0, 2.4, 3.0$. [2, 115] characterize the performance of look ahead for power-law topologies. For $\lambda = 3.0$ the largest degree in 1M nodes graph is less than 100, which brings the parameters close to real peer-to-peer networks.

Bimodal topologies. Along the lines of Section 6.2 we assume that there are two types of nodes in the network. Few nodes are connected to a large number of other nodes and, thus, they are very important for the operation of the network. Such nodes, which are typically called ultra-peers, are connected to the Internet through high-speed links and thus they can afford to connect to have many neighbors. The rest of the users have few neighbors. In our experiments in a 250K node graph we have used 500 nodes of degree 500.

Clustered topologies. We assume that there are clusters of users with very good connectivity inside

each cluster. The number of links between clusters are limited. In particular, in most of the experiments, we assume that the network is composed of a small number of clusters, and each cluster is a 3-regular random graph. Typical values are networks of 20 clusters of 10K nodes each, and 100 random connections between each pair of clusters.

6.6.2 Normalized Flooding

In this section we verify the results of Section 6.3 and compare the performance of normalized flooding to standard flooding. Both schemes perform similarly in regular topologies. However, when the topology contains nodes with high degrees, which is common in large unstructured communication networks, the normalization allows flooding to scale better.

With normalization it is easier to control the number of nodes reached by the flooding, thus, allowing, searching at a finer granularity (but still the number of nodes increases exponential with the initial TTL). In Table 15-A we give the mean number of unique peers discovered as a function of the initial time-to-live for four topologies of 250K nodes. We observe that in regular topologies standard flooding and normalized flooding behave similarly. In the other topologies however, which contain nodes of high degree, the increase in the number of peers is very fast (verifying Theorem 5); after the search reaches a high degree node, increasing the TTL by 1 or 2 will result in discovering a large part of the network. This tremendous increase, however, comes at the cost of reduced efficiency in the searching process (see Table 15-B). A large number of messages reach already discovered nodes. On the other hand, with normalized flooding (Theorem 6) we have more control on the number of messages generated and higher searching efficiency. Indeed, even in topologies with nodes of high degree, normalized flooding behaves similarly to searching in regular graphs.

Normalized flooding behaves better than standard flooding with respect to other metrics that are not shown in Table 15. For example the standard deviation (normalized by dividing with the mean) is much smaller (up to 4 times) with normalized flooding (observe that the standard deviation is of interest in the cases that the searching has not reached all the nodes of the network for all starting nodes). This indicates that the performance is more concentrated around the mean, and, thus, is less dependent of the starting node. Similar observations exist for other metrics of interest, like the minimum number of nodes discovered.

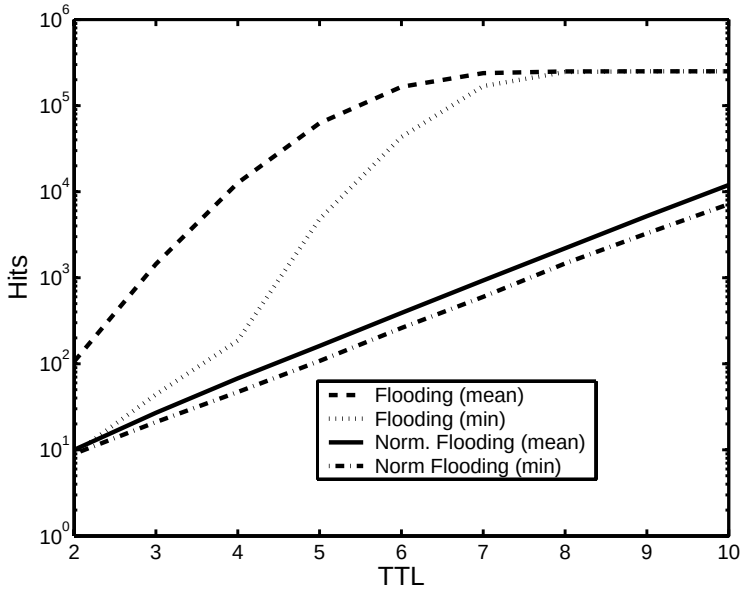


Figure 25: Number of unique peers discovered as a function of the initial time-to-live. In the case of normalized flooding the number of unique peers increases exponentially with the TTL, and, moreover, the increase is predictable and consistent for all nodes. In the case of regular flooding, the increase is much faster and depends on the node that initiates the flooding.

Moreover, the number of peers discovered as a function of the initial time-to-live follows a more predictable behavior; in contrast, without normalization there is a sharp jump in the number of peers discovered after few high-degree nodes are discovered. In Figure 25 we plot the number of nodes visited by the flooding as a function of the initial time-to-live. The straight line in the case of normalized flooding indicates that the horizon of the search increases exponentially. When searching in regular graphs we observe similar behavior. With flooding without normalization, the increase in topologies with nodes of high degrees is faster than exponential.

Observe that in order to discover the same number of nodes with normalized flooding as with standard flooding, we need to increase the TTL. Since the increase of the horizon is more predictable and can be roughly computed by either knowing the properties of the topology or by measuring the increase in the horizon when the TTL increases, it is easy to configure the value of TTL in order to reach at least a pre-specified number of nodes. Despite the fact that the increase in the horizon and in the messages is more predictable, it is still exponential. Later, we describe searching methods that allow finer granularity in controlling the number of messages.

Table 15: Performance and efficiency of flooding and normalized flooding.

A. Average number of unique peers discovered (F: Flooding, NF: Normalized Flooding).

TTL	Regular		Bimodal		Power-Law $\lambda = 1.4$		Power-Law $\lambda = 2.2$	
	F	NF	F	NF	F	NF	F	NF
2	9	9	370.3	9.8	28,216.5	10.9	106.1	10.0
3	21	21	28,463.5	24.9	177,192.9	32.1	1,441.9	27.0
4	45	45	126,581.9	58.5	247,700.5	87.4	12,633.6	67.7
5	93	93	212,759.4	133.6	249,993.2	231.9	62,480.5	161.4
6	188.9	188.9	244,392.6	297.2	249,999	610.6	165,543.6	389.6
7	380.7	380.7	249,635.0	661.7	249,999	1,594.1	238,522.0	930.1
8	763.9	763.9	249,991.4	1,422.8	249,999	4,122.4	249,807.9	2,197.1
9	1,528.7	1,528.7	249,999	3,014.8	249,999	10,430.9	249,999	5,194.6
10	3,051.0	3,051.0	249,999	6,212.4	249,999	24,599.3	249,999	11,953.9

B. Average efficiency of searching (unique peers over messages).

TTL	Regular		Bimodal		Power-Law $\tau = 1.4$		Power-Law $\tau = 2.2$	
	F	NF	F	NF	F	NF	F	NF
2	1.00	1.00	1.00	1.00	0.97	0.99	1.00	1.00
3	1.00	1.00	0.84	1.00	0.42	0.99	1.00	1.00
4	1.00	1.00	0.63	1.00	0.13	0.98	0.96	1.00
5	1.00	1.00	0.46	0.99	0.11	0.96	0.82	1.00
6	1.00	1.00	0.37	0.98	0.11	0.94	0.56	1.00
7	1.00	1.00	0.34	0.96	0.11	0.92	0.33	1.00
8	1.00	1.00	0.34	0.93	0.11	0.88	0.25	0.99
9	1.00	1.00	0.34	0.87	0.11	0.82	0.25	0.98
10	1.00	1.00	0.34	0.82	0.11	0.74	0.25	0.96

Note: Observe the more gradual increase in the number of peers discovered and the more gradual and slow decrease of efficiency when normalized flooding is used. This indicates better granularity when using normalized flooding compared to using regular flooding.

6.6.3 Evaluation of 1-step replication

Replication of one step is a practical way to improve the performance of searching by allowing each node to answer queries on behalf of its neighbors. This advantage comes at the cost of replicating information about the content of the neighbors, but, this cost is paid once when a new neighbor arrives and is amortized over a large number of messages that go through the node. This scheme is particularly effective when there are nodes of high degree in the network, as explained in Section 6.4. Visiting the nodes of large degree, and at the same time their neighbors, guarantees that a large portion of the network has been covered.

The performance of searching in networks of 250K nodes with normalized flooding and with random walks both with and without 1-step replication is given in Table 16. This table validates Theorems 8 and 9. In Table 16 we report the average number of unique peers discovered, but similar observations are given for the other statistics, like the standard deviation and the minimum number of peers discovered.

Compare Table 16 with Table 15. In the case of regular graphs normalized flooding with 1-step replication and TTL τ behaves very similarly to normalized flooding with TTL $\tau + 1$. In regular graphs random walks with 1-step replication perform better than floodings by more than 20%.

The advantage of 1-step replication becomes clear in graphs with large degrees, like the power-law graphs and the bimodal graph of Table 16. The number of hits significantly improved compared to not using 1-step replication (Table 15). The main reason is that both methods, flooding and random walks, quickly discover the nodes of high degree and through them discover a large portion of the nodes in the network. In the cases of graphs with large degrees the performance of normalized flooding is better than random walk (in the worst case by approximately 20%); on the other hand, in topologies with clusters, random walks perform slightly better.

In all cases the preprocessing to implement 1-step replication is given in the last line of Table 16. This cost is amortized over a large number of queries.

6.6.4 Evaluation of random walk with lookahead

An extension of the previous scheme is to perform a short random walk with shallow local floodings, of TTL $\tau = 2$ on every step (or, every few steps). We call this random walk with lookahead. Since

Table 16: Performance of searching with 1-step replication.

A. Normalized Flooding

TTL	Regular	Bimodal	Power-Law 1.4	Power-Law 2.2	Clustered
2	21	1,150.4	30,219.5	162.3	32.9
3	45	2,580.5	56,629.5	419.1	74.3
4	93.0	5,820.5	88,795.1	1,122.1	162.1
5	188.9	12,256.9	112,658.4	2,605.3	345.7
6	380.7	25,594.8	132,352.9	5,918.2	738.6
7	763.9	49,742.8	152,177.2	12,059.1	1,578.9
8	1,528.7	87,841.0	176,396.6	23,414.3	3,211.6
9	3,051.0	125,520.9	202,415.0	43,747.4	6,329.7
10	6,068.1	146,710.3	224,319.0	79,205.6	11,962.3
Pre-processing	750,000	990,852	2,446,674	1,265,536	1,019,500

B. Random Walks

TTL	Regular	Bimodal	Power-Law 1.4	Power-Law 2.2	Clustered
2	30.1	813.0	26,649.7	133.4	25.2
3	69.1	1,905.0	55,776.0	407.5	55.3
4	148.2	4,030.1	91,807.4	972.3	127.4
5	304.1	8,730.9	113,002.1	2,190.5	270.5
6	616.3	18,045.4	131,526.0	5,214.3	614.8
7	1,239.6	36,256.0	151,263.1	10,746.3	1,347.0
8	2,484.6	67,627.9	174,907.1	20,442.5	2,961.0
9	4,948.0	108,043.0	199,615.3	38,695.5	6,388.4
10	9,806.2	139,712.5	222,433.2	69,877.1	13,678.1
Pre-processing	750,000	990,852	2,446,674	1,265,536	1,019,500

Note: (1) In the case of Random Walks (Table B) the TTL column indicates that the random walker is using the same number of messages as if performing normalized flooding with that TTL. (2) Comparing to Table 15, we observe that 1-step replication increased the performance of searching via normalized flooding substantially for the same number of messages.

Table 17: Performance topologies of 1M nodes.

A. Power-law graph with $\alpha = 2.0$.

Metric	Flooding	Random-Walk	RW-look ahead $\tau = 2$
Median	12,652.50	24,692.50	26,097.00
Mean	27,773.31	24,694.32	28,616.97
Min	281	24,457	13,859
Max	393,040	24,930	71,167
Std	42,876.55	87.17	10,605.26
20-th perc.	996	24,495	14,972
Messages	32,640	32,640	32,851
TTL	4	-	2
Time	4	32,640	430

B. 6-Regular random graph.

Metric	Flooding	Random-Walk	RW-look ahead $\tau = 2$
Median	17,668.50	17,620.00	16,285.50
Mean	17,395.52	17,620.13	16,282.10
Min	10,576	17,389	15,519
Max	19,989	17,816	17,098
Std	1,477.91	73.48	296.20
20-th perc	13,238	17,453	15,667
Messages	22,386	22,386	22,656
TTL	6	-	2
Time	6	22,386	2,152

Note: In the RW-look ahead method, the number of samples between successive floodings is 4.

we do not think that it is realistic to maintain replicas of your neighbors' neighbors, therefore we charge the algorithm for all the messages generated by both the random walk and the local flooding. In Table 17 we show the performance of random walk with lookahead $\tau = 2$ in a regular graph and in power-law graph with few large degrees. The main observation is that the performance of the random walk with lookahead is similar in terms of unique peers discovered to performing a long random walk without lookahead, but, the response time is much smaller. (See also [106] for a different application of lookahead 2.)

6.6.5 Edge criticality and searching with weights

In Table 18 we give some statistics of the performance of generalized searching in topologies with clusters. This is experimental validation of Section 6.5. We study the performance of generalized

Table 18: Performance of generalized searching for various assignments of edge criticality.

Method	Min	Mean	Median	Max	Std	Mean Response Time
Proportional	12,869	17,276.30	17,236.0	21,523	1,453.9	8.1
Quadratic	16,054	21,036.31	21,194.0	23,143	1,061.6	10.9
Cubic	18,236	22,248.79	22,386.0	23,292	623.8	14.0
Fifth power	20,037	22,398.52	22,453.0	22,997	296.1	20.7
Tenth power	19,581	21,494.50	21,515.5	22,049	236.8	36.7
Flooding with TTL 6	10,410	16,377.83	15,595.5	30,859	3,223.2	6
Random Walk	20,034	20,328.72	20,330.0	20,591	84.8	28,026
Uniform Sampling	24,252	24,439.99	24.441.5	24,615	50.4	1

Note: The average number of messages in all cases is equal to 28,026. Two links l_1 and l_2 receive budget proportional to the ratio of the number of shortest hop paths going through them in the case of proportional sharing. In the cases of quadratic, cubic, and tenth power the allocation of the budget is proportional to the second, the third, and the tenth power of the ratio respectively.

searching for different methods of assigning the weights on the edges. The weights depend on the number of the shortest hop paths that go through each link. Recall that we do not compute all the shortest paths, but, instead, sample by computing shortest-paths for a subset of the nodes (2% of the nodes). Since it is possible that the sampling does not cover all edges, we assign a value of one to each uncovered edge. The relative weight that a node assigns to its incident edges depends on the number of shortest-hop paths that use these edges. Assuming that r is this ratio, then, in Table 18, we experiment with different assignments of the weights that are proportional to various powers of that ratio r^i. By increasing the power i, we increase the priority we give to the critical edges, and, as indicated in the table, the performance of searching increases. Increasing the power biases the searching towards the direction that leads to the boundary nodes and, subsequently, to other clusters. Of course, increasing the power i beyond a certain point reduces the performance (the process degenerates into oscillations between clusters).

In regular graphs without clustering, generalized searching performs similarly to standard flooding.

Does generalized searching perform well to other topologies with good connectivity, like for example power-law graphs and bimodal topologies? In other words, could the assignment of edge criticality ever become harmful? In power-law graphs and bimodal topologies edges incident to large degree vertices carry a lot of shortest paths and, hence, are assigned large criticality. However

these edges do not belong to bad cuts. Therefore, in non-regular topologies we normalize the edge criticality by dividing the number of paths going through an edge (u, v) by the maximum of the degrees of u and v. We have observed experimentally that using normalized edge criticality makes generalized flooding behave very similarly to flooding. In other words, if we are careful about the normalization, generalized searching does not decrease the performance of known algorithms.

CHAPTER VII

CONCLUSION

The main purpose of the work is to relate basic network performance metrics to structural characteristics of underlying network topologies and to develop protocols that reinforce and exploit desired structural characteristics.

To that extend we have proposed the use of the fundamental graph theoretic properties of conductance and eigenvalues to measure the good connectivity of network topologies. We have also proposed adaptations of the basic properties that capture extra semantics. Our proposed methodology captures global network connectivity characteristics and can be related directly to the performance of basic network communication tasks.

We have defined a canonical problem that relates to the capability of the network topology to route efficiently traffic and have shown, by bounding conductance, that routing with low congestion is possible as the size of the network increases in typical communication topologies. Thus, even though communication networks grow in a decentralized way without global coordination, the resulting topology has good global connectivity.

We have shown that typical network topologies, like the Internet at the autonomous system level, have clusters of nodes with natural business and geographical proximity, and, moreover, the existence of clusters can be identified using spectral analysis. We have also demonstrated that such clusters affect the performance of routing.

In the case of peer-to-peer systems we propose topology maintenance heuristics that result in topologies with provable good connectivity. We have also related the performance of searching via random walks in networks with good connectivity to the statistical properties of random walks on expander graphs and, hence, gave analytical justifications for the use of random walks in peer-to-peer systems. We also demonstrated cases of practical interest in which random walks perform better than the currently used searching by flooding.

We have also proposed new searching primitives that combine floodings and random walks and

which take advantage of local information to improve the performance of searching. Our schemes have similar statistical properties to searching via random walks and, moreover, have low finish times as in the case of searching by flooding. The use of local topological and statistical information to improve global performance is a useful approach for various network communication problems. We plan to expand our understanding on how such information can be gathered and used to further improve the performance of searching and of topology construction.

REFERENCES

[1] ADAMIC, L., "Zipf, power-laws, and pareto, a ranking tutorial." <http://www.hpl.hp.com/research/idl/papers/ranking/>, 2002.

[2] ADAMIC, L. A., LUKOSE, R. M., HUBERMAN, B., and PUNIYANI, A. R., "Search in power-law networks," *Physical Review E.*, vol. 64, no. 046135, 2001.

[3] ADAMIC, L. A., LUKOSE, R. M., and HUBERMAN, B. A., "Local search in unstructured networks," in *Handbook of Graphs and Networks* (BORNHOLDT, S. and SCHUSTER, H. G., eds.), pp. 295–317, Wiley-VCH, 1st ed., 2003.

[4] AGARWAL, S., SUBRAMANIAN, L., REXFORD, J., and KATZ, R. H., "Characterizing the Internet hierarchy from multiple vantage points (data)." <http://www.cs.berkeley.edu/~sagarwal/research/BGP-hierarchy/>, 2004.

[5] AIELLO, W., CHUNG, F. R. K., and LU, L., "A random graph model for massive graphs," in *ACM Symposium on Theory of Computing*, pp. 171–180, 2000.

[6] AIELLO, W., CHUNG, F. R. K., and LU, L., "Random evolution in massive graphs," in *Proc. 42nd Symposium on Foundations of Computer Science (FOCS)*, pp. 510–519, IEEE, 2001.

[7] AJTAI, M., KOMLOS, J., and SZEMEREDI, E., "Deterministic simulation in logspace," in *Proc. 19th ACM Symp. on Theory of Computing (STOC)*, (New York, New York, United States), pp. 132–140, ACM Press, 1987.

[8] AKELLA, A., CHAWLA, S., KANNAN, A., and SESHAN, S., "On the scaling of congestion in the Internet graph," *SIGCOMM Computer Communication Review*, vol. 34, no. 3, pp. 43–56, 2004.

[9] ALDERSON, D., DOYLE, J., GOVINDAN, R., and WILLINGER, W., "Toward an optimization-driven framework for designing and generating realistic Internet topologies," in *ACM HotsNet-I*, (Princeton, NJ, USA), 2002.

[10] ALDOUS, D., "On the markov chain simulation method for uniform combinatorial distributions and simulated annealing," *Probab. Engrg. Inform. Sci.*, vol. 1, pp. 33–46, 1987.

[11] ALDOUS, D. and FILL, J., "Reversible markov chains and random walks on graphs," 2002. http://stat-www.berkeley.edu/users/aldous/RWG/book.html.

[12] ALON, N., "Eigenvalues and expanders," *Combinatorica*, vol. 6, no. 2, pp. 83–96, 1986.

[13] ALON, N. and SPENCER, J. H., *The probabilistic method*. Wiley-Interscience series in discrete mathematics and optimization, New York: Wiley, 2nd ed., 2000. Noga Alon, Joel H. Spencer. 25 cm. "A Wiley-Interscience publication.".

[14] AZAR, Y., FIAT, A., KARLIN, A. R., MCSHERRY, F., and SAIA, J., "Spectral analysis of data," in *ACM Symposium on Theory of Computing*, (Hersonissos, Greece), pp. 619–626, 2001.

[15] BARABASI, A.-L. and ALBERT, R., "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.

[16] BAWA, M., COOPER, B., CRESPO, A., DASWANI, N., GANESAN, P., GARCIA-MOLINA, H., KAMVAR, S., MARTI, S., SCHLOSSER, M., SUN, P., VINOGRAD, P., and YANG, B., "Peer-to-peer research at stanford," *ACM-SIGMOD*, vol. 32, no. 3, pp. 23–26, 2003.

[17] BAWA, M., GARCIA-MOLINA, H., GIONIS, A., and MOTWANI, R., "Estimating aggregates on a peer-to-peer network," Tech. Rep. 24, Stanford, April 2003.

[18] BERGE, C., *Graphs and hypergraphs*. Amsterdam, New York,: North-Holland Pub. Co.; American Elsevier Pub. Co., [rev. ed., 1973. Translated by Edward Minieka. illus. 24 cm. North-Holland mathematical library, v. 6 Original ed., 1970, has title: Graphes et hypergraphes.

[19] BOLLOBáS, B., RIORDAN, O., SPENCER, J., and TUSNáDY, G., "The degree sequence of a scale-free random graph process," *Random Structures and Algorithms*, vol. 18, no. 3, pp. 279–290, 2001.

[20] BOLLOBAS, B., *Random graphs*. Cambridge studies in advanced mathematics ; 73, Cambridge ; New York: Cambridge University Press, 2nd ed., 2001. Bela Bollobas. ill. ; 24 cm.

[21] BOLLOBAS, B. and VEGA, F. D. L., "The diameter of random regular graphs," *Combinatorica*, vol. 2, no. 2, pp. 125–134, 1982.

[22] BOLLOBS, B. and RIORDAN, O., "The diameter of a scale-free random graph," *Combinatorica*, vol. 4, pp. 5–34, 2004.

[23] BOYD, S., DIACONIS, P., and XIAO, L., "Fastest mixing markov chain on a graph," *SIAM Review*, vol. 46, no. 4, pp. 667–689, 2004.

[24] BRODER, A. and KARLIN, A., "Bounds on the cover time," *J. Theoretical Probab.*, vol. 2, pp. 110–120, 1989.

[25] BRODER, A., KUMAR, R., MAGHOUL, F., RAGHAVAN, P., RAJAGOPALAN, S., STATA, R., TOMIKNS, A., and WIENER, J., "Graph structure in the web," *Computer Networks*, vol. 33, no. 1–6, pp. 309–320, 2000.

[26] BROIDO, A. and CLAFFY, K., "Internet topology: connectivity of IP graphs," in *SPIE International Symposium on Convergence of IT and Communication*, 2001.

[27] BU, T. and TOWSLEY, D., "On distinguishing between Internet power law topology generators," in *Infocom*, 2002.

[28] CAIDA, "Netgeo tool," 2002.

[29] "Cooperative Association for Internet Data Analysis." <`http://www.caida.org`>, 2004.

[30] CALVERT, K., DOAR, M., and ZEGURA, E., "Modeling Internet topology," *IEEE Communications Magazine*, 1997.

[31] CARLSON, J. and DOYLE, J., "Highly optimized tolerance: A mechanism for powerlaws in design systems," *Physics review E*, vol. 60, no. 2, pp. 1412–1427, 1999.

[32] CHAWATHE, Y., RATNASAMY, S., BRESLAU, L., LANHAM, N., and SHENKER, S., "Making gnutella-like networks scalable," in *SIGCOMM*, (Karlsruhe, Germany), pp. 407–418, ACM, 2003.

[33] CHEN, Q., CHANG, H., GOVINDAN, R., JAMIN, S., SHENKER, S., and WILLINGER, W., "The origins of power-laws in Internet topologies revisited," *Infocom*, 2002.

[34] CHUNG, F. and LU, L., "The average distance in a random graph with given expected degrees," *Internet Mathematics*, vol. 1, no. 1, pp. 91–114, 2003.

[35] CHUNG, F., LU, L., and VU, V., "Eigenvalues of random power law graphs," *Annals of Combinatorics*, vol. 7, pp. 21–33, 2003.

[36] CHUNG, F., LU, L., and VU, V., "The spectra of random graphs with given expected degrees," *Proceedings of National Academy of Sciences*, vol. 100, no. 11, pp. 6313–6318, 2003. Extended version `http://www.math.ucsd.edu/~fan/wp/specp.pdf`.

[37] CHUNG, F. R. K., *Spectral graph theory*. Regional conference series in mathematics, no. 92, Providence, R.I.: Published for the Conference Board of the mathematical sciences by the American Mathematical Society, 1997. Fan R.K. Chung. 26 cm. "CBMS Conference on Recent Advances in Spectral Graph Theory held at California State University at Fresno, June 6-10, 1994"–T.p. verso.

[38] CHUNG, F. and LU, L., "Connected components in random graphs with given degree sequences," *Annals of Combinatorics*, vol. 6, pp. 125–145, 2002.

[39] COATES, M., HERO, A., NOWAK, R., and YU, B., "Internet tomography," *IEEE Signal Processing Magazine*, vol. 19, no. 3, pp. 47–65, 2002.

[40] COHEN, E., FIAT, A., and KAPLAN, H., "Associative search in peer to peer networks: Harnessing latent semantics," in *IEEE Infocom*, (San Francisco, CA, USA), 2003.

[41] COHEN, E. and SHENKER, S., "Replication strategies in unstructured peer-to-peer networks," in *ACM SigComm*, (Pittsburgh, PA, USA), 2002.

[42] COOPER, C. and FRIEZE, A., "On a general model for web graphs," in *Proceedings of ESA*, pp. 500–511, 2001.

[43] COOPER, C. and FRIEZE, A., "The cover time of sparse random graphs," in *Symposium on Discrete Algorithms (SODA)*, (Baltimore, Maryland), pp. 140 – 147, SIAM/ACM, 2003.

[44] COOPER, C. and FRIEZE, A., "The cover time of random regular graphs," 2004.

[45] COOPER, C. and FRIEZE, A., "The cover time of the preferential attachment graph," 2004.

[46] CVETKOVIAC, D. M., DOOB, M., and SACHS, H., *Spectra of graphs : theory and application*. Berlin: Deutscher Verlag der Wissenschaften, 1980. by Dragos M. Cvetkovic, Michael Doob, Horst Sachs. ill. ; 24 cm. Includes indexes.

[47] CVETKOVIAC, D. M., ROWLINSON, P., and SIMIAC, S., *Eigenspaces of graphs*. Encyclopedia of mathematics and its applications ; v. 66, Cambridge ; New York: Cambridge University Press, 1997. D. Cvetkovic, P. Rowlinson, S. Simic. ill. ; 24 cm.

[48] DILL, S., KUMAR, R., MCCURLEY, K., RAJAGOPOLAN, S., SIVAKUMAR, D., and TOMKINS, A., "Self-similarity in the web," in *International Conference on Very Large Data Bases*, (Rome), pp. 69–78, 2001.

[49] DOYLE, J. C., CARLSON, J., LOW, S. H., PAGANINI, F., VINNICOMBE, G., WILLINGER, W., HICKEY, J., PARRILO, P., and VANDENBERGHE, L., "Robustness and the internet: Theoretical foundations." <`http://netlab.caltech.edu/internet/`>, 2002.

[50] "eMule-project.net - Official eMule site." <`http://www.emule-project.net/`>, 2004.

[51] FABRIKANT, A., KOUTSOUPIAS, E., and PAPADIMITRIOU, C. H., "Heuristically optimized tradeoffs: A new paradigm for powerlaws in the Internet.," in *ICALP*, 2002.

[52] FALOUTSOS, M., FALOUTSOS, P., and FALOUTSOS, C., "On power-law relationships of the Internet topology," in *Proc. of ACM SIGCOMM*, 1999.

[53] FARKAS, I. J., DERENYI, I., BARABASI, A.-L., and VICSEK, T., "Spectra of real-world graphs: Beyond the semicircle law," *Physical Review E*, vol. 64, no. 026704, 2001.

[54] FELLER, W., *An introduction to probability theory and its applications*. Wiley series in probability and mathematical statistics, New York,: Wiley, 3d ed., 1967. illus. 24 cm.

[55] FESSANT, F. L., HANDURUKANDE, S., KERMARREC, A.-M., and MASSOULI, L., "Clustering in peer-to-peer file sharing workloads.," in *International Workshop on Peer-to-peer systems (IPTPS 04)*, (San Diego, CA, USA), 2004.

[56] FLAXMAN, A., FRIEZE, A. M., and FENNER:, T. I., "High degree vertices and eigenvalues in the preferential attachment graph," in *7th International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM) and 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, (Princeton, NY, USA), pp. 264–274, 2003.

[57] FRIEZE, A. M., "Disjoint paths in expander graphs via random walks: A short survey," in *Random '98*, Lecture Nodes in Computer Science (1518), pp. 1–14, Springer, 1998.

[58] GABBER, O. and GALIL, Z., "Explicit constructions of linear-sized superconcentrators," *Journal of Computer and System Sciences*, vol. 22, pp. 407–420, 1981.

[59] GAO, L., "On inferring autonomous system relationships in the Internet," in *IEEE Glogal Internet*, 2000.

[60] GIBSON, D., KLEINBERG, J. M., and RAGHAVAN, P., "Inferring web communities from link topology," *UK Conference on Hypertext*, 1998.

[61] GILLMAN, D., "A chernoff bound for random walks on expander graphs," *Journal on Computing*, vol. 27, no. 4, pp. 1203–1220, 1998.

[62] GKANTSIDIS, C., MIHAIL, M., and SABERI, A., "Conductance and congestion in power law graphs," in *ACM SIGMETRICS*, (San Diego, CA, US), pp. 148–159, ACM Press, 2003.

[63] GKANTSIDIS, C., MIHAIL, M., and SABERI, A., "Random walks in peer-to-peer networks," in *IEEE Infocom*, (Hong Kong), 2004.

[64] GKANTSIDIS, C., MIHAIL, M., and SABERI, A., "Hybrid search schemes for unstructured peer-to-peer networks," in *IEEE Infocom*, 2005.

[65] GKANTSIDIS, C., MIHAIL, M., and ZEGURA, E., "The markov chain simulation method for generating connected power law random graphs," in *SIAM Alenex*, (Baltimore, MD), 2003.

[66] GKANTSIDIS, C., MIHAIL, M., and ZEGURA, E., "Spectral analysis of Internet topologies," in *IEEE Infocom*, (San Francisco, CA, US), 2003.

[67] "Gnutella." <http://p2pjournal.com/main/gnutella.htm>.

[68] GOH, K.-I., KAHNG, B., and KIM, D., "Spectra and eigenvectors of scale-free networks," *Physical Review E.*, vol. 64, no. 051903, 2001.

[69] GOLUB, G. H. and LOAD, C. F. V., "Matrix computations," *Johns Hopkins University Press*, 1989.

[70] HUSBANDS, P., SIMON, H., and DING, C., "On the use of the singular value decomposition for text retrieval," in *1st SIAM Computational Information Retrieval Workshop*, 2000.

[71] IAMNITCHI, A., RIPEANU, M., and FOSTER, I., "Small-world file-sharing communities," in *IEEE Infocom*, (Hong Kong), 2004.

[72] IMPAGLIAZZO, R. and ZUCKERMAN, D., "How to recycle random bits," in *30th IEEE Symposium on the Foundations of Computer Science*, (Research Triangle Park, NC), pp. 248–253, IEEE, 1989.

[73] JIN, C., CHEN, Q., and JAMIN, S., "Inet: Internet topology generator," Tech. Rep. CSE-TR-433-00, University of Michigan, 2000.

[74] JOVANOVIC, M., ANNEXSTEIN, F., and BERMAN, K., "Modeling peer-to-peer network topologies through small-world models and power laws," in *IX Telecommunications Forum*, 2001.

[75] KAHALE, N., "Eigenvalues and expansion of regular graphs," *Journal of the Association for Computing Machinery*, vol. 42, no. 5, pp. 1091–1105, 1995.

[76] KAN, G., "How gnutella happened." <http://news.dmusic.com/print/3641>, 2001.

[77] KARAGIANNIS, T., BROIDO, A., BROWNLEE, N., CLAFFY, K., and FALOUTSOS, M., "Is p2p dying or just hiding?," in *IEEE Globecom*, (Dallas, Texas, USA), 2004.

[78] "Kazaa." <http://www.kazaa.com>, 2004.

[79] KEMPE, D., DOBRA, A., and GEHRE, J., "Gossip-based computation and aggregate information," in *IEEE FOCS'03*, pp. 482–491, IEEE, 2003.

[80] KEMPE, D. and MCSHERRY, F., "A decentralized algorithm for spectral analysis," in *Proc. the 36th annual ACM symposium on Theory of computing*, (Chicago, IL, USA), pp. 561 – 568, ACM, 2004.

[81] KERMARREC, A.-M., "Self-clustering in peer-to-peer overlays," in *International Workshop on Self-* Properties in Complex Information Systems*, (Bertinoro, Italy), pp. 89–92, 2004.

[82] KERNER, T., "Public route server and looking glass site list." <`http://www.traceroute.org`>, 2004.

[83] KLEIN, P., STEIN, C., and TARDOS, ., "Leighton-rao might be practical: faster approximation algorithms for concurrent flow with uniform capacities," in *Annual ACM Symposium on Theory of Computing (STOC)*, (Baltimore, Maryland, United States), pp. 310–321, 1990.

[84] KLEIN, P., PLOTKIN, S., STEIN, C., and TARDOS, V., "Faster approximation algorithms for the unit capacity concurrent flow problem with applications to routing and finding sparse cuts," *Journal on Computing*, vol. 23, no. 3, pp. 466–487, 1994.

[85] KLEINBERG, J., "Navigation in a small world," *Nature*, vol. 406, 2000.

[86] KLEINBERG, J., "Hubs, authorities, and communities on the WWW," *ACM Computing Surveys*, vol. 31, no. 4es, 1999.

[87] KLEINBERG, J. and RUBINFELD, R., "Short paths in expander graphs," in *Proc. 37th IEEE Symposium on Foundations of Computer Science*, IEEE, 1996.

[88] KUMAR, A., MERUGU, S., XU, J., and YU, X., "Ulysses: A robust, low-diameter, low-latency peer-to-peer network," in *IEEE ICNP 2003*, (Atlanta, GA, USA), 2003.

[89] KUMAR, R., RAGHAVAN, P., RAJAGOPALAN, S., D., S., TOMKINS, A., and UPFAL, E., "Stochastic models for the web graphs," in *Proc. 41st Symposium on Foundations of Computer Science (FOCS)*, pp. 57–65, IEEE, 2000.

[90] KUMAR, R., RAJAGOPALAN, S., SIVAKUMAR, D., and TOMKINS, A., "Crawling the web for emerging cyber-communities," *WWW8/Computer Networks*, vol. 31, no. 11–16, pp. 1481–1493, 1999.

[91] LAKHINA, A., BYERS, J., CROVELLA, M., and MATTA, I., "On the geographical location of Internet resources," *IEEE J-SAC, Special Issue on Internet and WWW Measurement, Mapping and Modeling*, vol. 21, no. 6, pp. 934–948, 2003.

[92] LAW, C. and SIU, K.-Y., "Distributed construction of random expander networks," in *IEEE Infocom*, (San Francisco, CA, USA), 2003.

[93] LEIBOWITZ, N., RIPEANU, M., and WIERZBICKI, A., "Deconstructing the KaZaA network," in *3rd IEEE Workshop on Internet Applications (WIAPP'03)*, (San Jose, CA, USA), 2003.

[94] LEIGHTON, F. T. and RAO, S., "An approximate max-flow min-cut theorem for uniform multicommodity flow problem with applications to approximation algorithms," in *29th IEEE Symp. on Foundations of Computer Science (FOCS'88)*, pp. 422–431, IEEE, 1988.

[95] LEIGHTON, F. T. and RAO, S., "Multicommodity maxflow min-cut theorems and their use in designing approximation algorithms," *Journal of the ACM*, vol. 46, no. 6, pp. 787–832, 1999.

[96] LEIGHTON, F. T., *Introduction to parallel algorithms and architectures : arrays, trees, hypercubes*. San Mateo, Calif.: M. Kaufmann Publishers, 1992. F. Thomson Leighton. ill. ; 24 cm.

[97] LEIGHTON, F. and RAO, S., "Circuit switching: Multicommodity flow based approach," in *Workshop on Randomized Parallel Computing*, 1996.

[98] LEIGHTON, T. and RAO, S., "Circuit switching: Multicommodity flow based approach," in *Workshop on Randomized Parallel Computing*, 1995.

[99] LI, Z. and MOHAPATRA, P., "The impact of topology on overlay routing service," in *IEEE Infocom*, (Hong Kong), 2004.

[100] LIMEWIRE.ORG, "Snapshots of the gnutella network." <`http://crawler.limewire.org/data.html`>, 2002.

[101] LINIAL, N., LONDON, E., and RABINOVICH, Y., "The geometry of graphs and some of its algorithmic applications.," *Combinatorica*, vol. 15, no. 2, pp. 215–245, 1995.

[102] LOVASZ, L., "Combinatorial problems and exercises," *North Holland, Amsterdam*, 1979.

[103] LUA, E. K., CROWCROFT, J., and PIAS, M., "Highways: Proximity clustering for scalable peer-to-peer network," in *4th International Conference on Peer-to-Peer Computing*, (Zurich, Switzerland), 2004.

[104] LUBOTZKY, A., PHILLIPS, R., and SARNAK, P., "Ramanujan graphs," *Combinatorica*, vol. 8, no. 3, pp. 261–277, 1988.

[105] LV, Q., CAO, P., COHEN, E., LI, K., and SHENKER, S., "Search and replication in unstructured peer-to-peer networks," in *International Conference on Supercomputing*, (New York, New York, USA), pp. 84–95, ACM Press, 2002. Extended version in `http://www.cs.princeton.edu/~qlv/download/searchp2p_full.pdf`.

[106] MANKU, G. S., NAOR, M., and WIEDER, U., "Know thy neighbor's neighbor: the power of lookahead in randomized p2p networks," in *ACM Symposium on Theory of Computing (STOC)*, (Chicago, IL, USA), pp. 54 – 63, ACM, 2004.

[107] MCSHERRY, F. and KEMPE, D., "A decentralized algorithm for spectral analysis," in *Proc. of STOC'04*, ACM Press, 2004.

[108] MEDINA, A., MATTA, I., and BYERS, J., "On the origin of power-laws in Internet topologies," *ACM Computer Communications Review*, 2000.

[109] MEDINA, A., MATTA, I., and BYERS, J., "BRITE: A flexible generator of Internet topologies," 2001.

[110] MERUGU, S. ., SRINIVASAN, S., and ZEGURA, E., "Adding structure to unstructured peer-to-peer networks: the use of small-world graphs," *Journal of Parallel and Distributed Computing*, 2004.

[111] "Bob Metcalfe eats his words," *Internet Computing Online*, vol. 1, no. 3, 1997. Also available at <`http://www.computer.org/internet/v1n3/eats9702.htm`>.

[112] METCALFE, B., *Internet Collapses and Other InfoWorld Punditry*. Hungry Minds, 1st ed., 2000.

[113] MIHAIL, M. and PAPADIMITRIOU, C., "On the eigenvalue power-law," in *RANDOM*, (Harvard, MA), 2002.

[114] MIHAIL, M., PAPADIMITRIOU, C., and SABERI, A., "On certain connectivity properties of the internet topology," in *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, (Washington, DC, USA), p. 28, IEEE Computer Society, 2003.

[115] MIHAIL, M., SABERI, A., and TETALI, P., "Random walks with lookahead in power law random graphs," 2004.

[116] MOLLOY, M. and REED, B., "A critical point for random graphs with a given degree sequence," *Random Structures and Algorithms*, vol. 6, pp. 161–180, 1995.

[117] MOLLOY, M. and REED, B., "The size of the largest component of a random graph on a fixed degree sequence," *Combinatorics, Probability and Computing*, vol. 7, pp. 295–306, 1998.

[118] MOTWANI, R. and RAGHAVAN, P., *Randomized algorithms*. Cambridge ; New York: Cambridge University Press, 1995. Rajeev Motwani, Prabhakar Raghavan. ill. ; 26 cm.

[119] "Mutella." <`http://mutella.sourceforge.net/`>, 2004.

[120] NEWMAN, M., "Assortative mixing in networks," *Phys. Rev. Lett.*, vol. 89, 2002.

[121] "Raw routing table information." <`http://moat.nlanr.net/Routing/rawdata/`>, 2000.

[122] ORAM, A., *Peer-to-peer : harnessing the benefits of a disruptive technology*. Beijing ; Sebastopol, CA: O'Reilly, 1st ed., 2001. edited by Andy Oram. Subtitle on spine: Harnessing the power of disruptive technologies ill. ; 24 cm.

[123] PADMANABHAN, V., QIU, L., and WANG, H., "Server-based inference of internet performance," in *IEEE Infocom*, 2003.

[124] PAGE, L., BRIN, S., MOTWANI, R., and WINOGRAD, T., "The pagerank citation ranking: Bringing order to the web," *Stanford Digital Library Technologies Project*, 1998.

[125] PANDURANGAN, G., RAGHAVAN, P., and UPFAL, E., "Building low-diameter p2p networks," in *42nd Annual Symposium on Foundations of Computer Science (FOCS01)*, pp. 492–499, 2001.

[126] PAPADIMITRIOU, C., RAGHAVAN, P., TAMAKI, H., and VEMPALA, S., "Latent semantic indexing: A probabilistic analysis," *Proc. Principles of Database Systems (PODS)*, 1999.

[127] PAPADIMITRIOU, C. H., *Computational complexity*. Reading, Mass.: Addison-Wesley, 1994. Christos H. Papadimitriou. ill. ; 25 cm.

[128] PATCH, K., "Simple search lightens net load." <`http://www.trnmag.com/Stories/2004/090804/Simple_search_lightens_Net_load_090804.html`>, 2004.

[129] PIPPENGER, N., "On rearrangeable and non-blocking switching networks," *Journal of Computer and System Sciences*, vol. 17, no. 2, pp. 145–162, 1978.

[130] PIPPINGER, N., "Superconcentrators," *SIAM Journal on Computing*, vol. 6, no. 2, pp. 298–304, 1977.

[131] PIPPINGER, N., "Information theory and the complexity of switching networks," in *Proc. 16th Symposium on Foundations of Computer Science (FOCS)*, pp. 113–118, 1978.

[132] POTHEN, A., SIMON, H. D., and LIOU, K.-P., "Partitioning sparse matrices with eigenvectors of graphs," *Matrix Analysis and Applications*, vol. 11, no. 3, pp. 430–452, 1990.

[133] POWERS, D. L., "Graph partitioning by eigenvectors," *Linear Algebra and its Applications*, vol. 101, no. 121-133, 1988.

[134] QUARTERMAN, J. S., "Imminent death of the Internet?," *Matrix News*, vol. 6, no. 6, 1996.

[135] RADOSLAVOV, P., TANGMUNARUNKIT, H., YU, H., GOVINDAN, R., SHENKER, S., and ESTRIN, D., "On characterizing network topologies and analyzing their impact on protocol design," Tech. Rep. USC-CS-TR-00-731, University of Southern California, March 2000 2000. Available at `http://citeseer.nj.nec.com/radoslavov00characterizing.html`.

[136] RAGHAVAN, P., "Information retrieval algorithms: A survey," *Proc. 8th SIAM Symposium on Discrete Algorithms (SODA)*, 1997.

[137] RATNASAMY, S., FRANCIS, P., HANDLEY, M., KARP, R., and SHENKER, S., "A scalable content-addressable network," in *ACM SigComm*, (San Diego, CA, USA), 2001.

[138] REHKTER, Y. and GROSS, P., "Application of the border gateway protocol in the internet," RFC 1655, IETF, July 1994.

[139] RIPEANU, M., FOSTER, I., and IAMNITCHI, A., "Mapping the Gnutella network: Properties of large-scale peer-to-peer systems and implications for system design," *IEEE Internet Computing Journal special issue on peer-to-peer networking*, vol. 6, no. 1, 2002.

[140] RITTER, J., "Why gnutella can't scale. no, really.." <`http://www.darkridge.com/~jpr5/doc/gnutella.html`>, 2001.

[141] ROWSTRON, A. and DRUSCHEL, P., "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," in *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, (Heidelberg, Germany), pp. 329–350, 2001.

[142] SARSHAR, N., BOYKIN, P. O., and ROYCHOWDHURY, V., "Scalable percolation search in power law networks," in *In Proceedings of the Fourth IEEE International Conference on Peer-to-Peer Computing (P2P2004)*, pp. 2–9, IEEE Press, 2004.

[143] SARSHAR, N., ROYCHOWDURY, V., and BOYKIN, P. O., "Percolation-based search on unstructured peer-to-peer networks," in *4th IEEE International Conference on Peer-to-Peer Computing*, (Zurich, Switzerland), 2004.

[144] SINCLAIR, A., *Algorithms for random generation and counting : a Markov chain approach*. Progress in theoretical computer science, Boston: Birkhauser, 1993. Alistair Sinclair. 25 cm. Rev. version of the author's thesis (Ph. D.)–University of Edinburgh, 1988.

[145] SPENCER, J. H., *Ten lectures on the probabilistic method*. CBMS-NSF regional conference series in applied mathematics ; 64, Philadelphia, Pa.: Society for Industrial and Applied Mathematics, 2nd ed., 1994. Joel Spencer. ill. ; 26 cm.

[146] STEWART, G. W. and SUN, J.-G., *Matrix perturbation theory*. Computer science and scientific computing, Boston: Academic Press, 1990. G.W. Stewart, Ji-guang Sun. 24 cm.

[147] STOICA, I., MORRIS, R., KARGER, D., KAASHOEK, M. F., and BALAKRISHNAN, H., "Chord: A scalable peer-to-peer lookup service for Internet applications," in *ACM SigComm*, (San Diego, CA, USA), 2001.

[148] SUBRAMANIAN, L., AGARWAL, S., REXFORD, J., and KATZ, R., "Characterizing the Internet hierarchy from multiple vantage points," in *IEEE Infocom*, 2002.

[149] TANG, C., XU, Z., and DWARKADAS, S., "Peer-to-peer information retrieval using self-organizing semantic overlay networks," in *ACM SigComm*, (Karlsruhe, Germany.), 2003.

[150] TANGMUNARUNKIT, H., GOVINDAN, R., JAMIN, S., SHENKER, S., and WILLINGER, W., "Network topology generators: Degreebased vs. structural," in *ACM SigComm*, (Pittsburgh, PA), ACM, 2002.

[151] TOWSLEY, D., "Modeling the internet: Seeing the forest through the trees," 2002. Keynote Address, Sigmetrics.

[152] TSOUMAKOS, D. and ROUSSOPOULOS, N., "Adaptive probabilistic search for peer-to-peer networks," in *3rd IEEE International Conference on P2P Computing*, (Linkoping, Sweden), 2003.

[153] VARDI, Y., "Network tomography: Estimating source-destination traffic intensities from link data," *Journal of the American Statistical Association*, vol. 91, no. 433, pp. 365–377, 1996.

[154] VAZIRANI, V. V., *Approximation algorithms*. Berlin ; New York: Springer, 2001. Vijay V. Vazirani. ill. ; 25 cm.

[155] VUKADINOVIC, D., HUANG, P., and ERLEBACH, T., "A spectral analysis of the Internet topology," *Dimacs Workshop on Internet and WWW Measurement, Mapping, and Modeling*, 2001. Also available at `citeseer.nj.nec.com/vukadinovic01spectral.html`.

[156] WAXMAN, B. M., "Routing of multipoint connections," *IEEE Jour. Selected Areas in Communications (Special Issue: Broadband Packet Communications)*, vol. 6, no. 9, pp. 1617–1622, 1988.

[157] WILKINSON, J. H., *The algebraic eigenvalue problem*. Oxford,: Clarendon Press, 1965. by J. H. Wilkinson. illus. 25 cm. Monographs on numerical analysis.

[158] WILLINGER, W. and DOYLE, J., "Robustness and the internet: Design and evolution." 22 Oct 2005 <`http://netlab.caltech.edu/internet/`>, 22 Oct 2005 2002.

[159] YOOK, S.-H., JEONG, H., and BARABASI, A.-L., "Modeling the Internet's large-scale topology," *Proceedings of the National Academy of Sciences*, vol. 99, no. 21, pp. 13382–13386, 2002.

[160] ZEGURA, E., CALVERT, K., and BHATTACHARJEE, S., "How to model an internetwork," in *IEEE INFOCOM*, 1996.

[161] ZEGURA, E., CALVERT, K., and DONAHOO, M., "A quantitative comparison of graph-based models for Internet topology," *Transactions on Networking, Vol 5, No 6*, pp. 770–783, 1997.

VITA

Christos Gkantsidis joined the Ph.D. program of the College of Computing of Georgia Institute of Technology in 1999. Prior to joining Georgia Tech, he earned a bachelors degree in Computer Science and Engineering from the University of Patras, Greece. His interest in computer networking developed during his undergraduate years, when he worked as a network administrator for the Computer Technology Institute, Patras, Greece. While in graduate school, his research interests drifted toward networking problems, especially content distribution, in the beginning of his Ph.D., and modeling and designing algorithms for complex communication networks, which is the focus of his Ph.D. thesis. While a graduate student, Christos had the opportunity to work as an intern for Sprint Labs and Microsoft Research.

Christos is a member of the IEEE and the ACM.